Faculty of Engineering and Technology

Department of Electronics and Communication Engineering

Jain Global Campus, Kanakapura Taluk - 562112
Ramanagara District, Karnataka, India

2018-2022

A Project Report on

"SMART ATTENDANCE SYSTEM USING FACE RECOGNITION"

Submitted in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

Bhavana. L

18BTREC010

E Ajith Bhargav

18BTREC018

G Sai Dharani

18BTREC019

N Lohithas Reddy

18BTREC046

Under the guidance of

Dr. R. Sukumar

HOD

Department of Electronics and Communication Engineering

Faculty of Engineering & Technology

JAIN DEEMED-TO-BE UNIVERSITY

Faculty of Engineering & Technology

Department of Electronics & Communication Engineering

Jain Global campus

Kanakapura Taluk - 562112

Ramanagara District

Karnataka, India

CERTIFICATE

This is to certify that the project work titled **"SMART ATTENDANCE SYSTEM USING FACE RECOGNITION"** is carried out by **Bhavana. L (18BTREC010), G Sai Dharani (18BTREC019), E Ajith Bhargav(18BTREC018), N Lohithas Reddy (18BTREC046),** are bonafide students of Bachelor of Technology at the Faculty of Engineering & Technology, JAIN DEEMED-TO-BE UNIVERSITY, Bengaluru in partial fulfillment for the award of degree in Bachelor of Technology in Electronics and Communication Engineering, during the academic year **2021-2022**.

Dr. R. Sukumar	**Dr. R. Sukumar**	**Dr. Hariprasad S.A**
HOD	Head of the Department, Electronics and Communication,	Director,
Dept. of ECE,	Faculty of Engineering & Technology,	Faculty of Engineering & Technology,
Faculty of Engineering & Technology,	JAIN DEEMED-TO-BE UNIVERSITY	JAIN DEEMED-TO-BE UNIVERSITY
JAIN DEEMED-TO-BE UNIVERSITY	Date:	Date:
Date:		

Name of the Examiner Signature of Examiner

1.

2.

DECLARATION

We, **Bhavana. L (18BTREC010), G Sai Dharani (18BTREC019), E Ajith Bhargav(18BTREC018), N Lohithas Reddy (18BTREC046),** are student's of eighth semester B.Tech in **Electronics and Communication Engineering**, at Faculty of Engineering & Technology, **JAIN DEEMED-TO-BE UNIVERSITY**, hereby declare that the project titled **"SMART ATTENDANCE SYSTEM USING FACE RECOGNITION"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Electronics and Communication Engineering** during the academic year **2021-2022**. Further, the matter presented in the project has not been submitted previously by anybody for the award of any degree or any diploma to any other University, to the best of our knowledge and faith.

Signature

Bhavana. L

18BTREC010

E Ajith Bhargav

18BTREC018

G Sai Dharani

18BTREC019

N Lohithas Reddy

18BTREC046

Place : Bengaluru

Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to **Faculty of Engineering & Technology, JAIN DEEMED-TO-BE UNIVERSITY** *for providing us with a great opportunity to pursue our Bachelor's Degree in this institution.*

In particular we would like to thank ***Dr. Hariprasad S.A, Director,*** **Faculty of Engineering & Technology**, **JAIN DEEMED-TO-BE UNIVERSITY** *for his constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Dr. R. Sukumar,*** **Head of the department**, **Electronics and communication Engineering**, **JAIN DEEMED-TO-BE UNIVERSITY,** *for providing right academic guidance that made our task possible.*

We would like to thank our guide ***Dr. R. Sulumar*** *HOD,* **Dept. of Electronics and Communication Engineering**, **JAIN DEEMED-TO-BE UNIVERSITY***, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

We would like to thank our Project Coordinator ***Mr. Sunil M P*** *and all the staff members of Electronics and Communication for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in completing the Project work successfully.

Signature of Students

Bhavana. L

E Ajith Bhargav

G Sai Dharani

N Lohithas Reddy

ABSTARCT

A face recognition system is one of the biometric information processes, its applicability is easier and working range is larger than others, i.e.; fingerprint, iris scanning, signature, etc. A face recognition system is designed, implemented and tested. The system uses a combination of techniques in two topics; face detection and recognition. The face detection is performed on live acquired images without any application field in mind. Processes utilized in the system are white balance correction, skin like region segmentation, facial feature extraction and face image extraction on a face candidate. Then a face classification method that uses Feedforward Neural Network is integrated in the system. The system is tested with a database generated in the laboratory with 26 people. The tested system has acceptable performance to recognize faces within intended limits. System is also capable of detecting and recognizing multiple faces in live acquired images.

TABLE OF CONTENTS

List of Figures viii

List of Tables vii

Nomenclature used 0

Chapter 1 **01**

1. INTRODUCTION **01**

1.1 Literature Survey 02

1.2 Limitations of the Current Work 04

1.3 Problem Statement 04

1.4 Objectives 05

1.5 Methodology 05

1.6 Hardware and Software tools used 07

Chapter 2

2. BASIC THEORY 08

Chapter 3 **10**

3. TOOL DESCRIPTION **10**

Chapter 4

4. IMPLEMENTATION **13**

4.1 Hardware Design and Implementation 13

4.1.1 Start with MATLAB 13

4.1.2 File extension in MATLAB 14

4.1.3 M files 15

4.1.4 Plotting using MATLAB 16

4.1.5 Serial communication using MATLAB 20

4.1.6 MATLAB command list 25

4.2 Software algorithm

Chapter 5 **38**

5. RESULTS AND DISCUSSION **38**

CONCLUSIONS AND FUTURE SCOPE **39**

REFERENCES viii

APPENDICES

APPENDIX – I **ix**

APPENDIX – II **x**

DETAILS OF PAPER PUBLICATION(ALONG WITH PAPER) **xi**

INFORMATION REGARDING STUDENTS **xii**

BATCH PHOTOGRAPH ALONG WITH GUIDE **xiii**

LIST OF FIGURES

Fig. No.	Description of the figure	Page no:
1.1	Enrolment phase	05
1.2	Feature extraction	06
4.1	Start with MATLAB	11
4.2	Editor window	11
4.3	File window	13
4.4	plotting window	14

LIST OF TABLES

Table No.	Description of the figure	Page no:
4.1	MATLAB command list	22

Chapter 1

1.Introduction

During this era of technology and automation we are still using the same old ways of attendance management. The most important thing in the classroom/office is attendance which is directly linked to the academic performance of the students/employees. Recently, some students are husky with better skills during Lectures only when there is massive classroom control (Research Gale, 2018).

The more efficient the attendance system the mare is participation and learning. In the past we were using techniques like roll numbering calling and signing against a particular roll number. These methods carry a high chance of proxy and are time consuming. We came across the idea of automating this process through modern day technologies to get a well maintained and disciplined attendance. Facial recognition systems along with suitable hardware and software will help meet the goals of this project.

Facial recognition system is derived incovalion of image processing Image processing deals with the extraction of needy data that can be related to digital Image and in technology advancement it plays a unique role. Our core focus will be on receiving digital Images and then making use of programs and algorithms to get useful Information out of it. As the pictorial information is for the image processing work on it and make it esetl human interpretation.

That information from image processing will play a great role and help in various walks of life where it could be implemented. The applications of image processing are vast and can be applied in most scenarios where imaging data could be related to pre-determined algorithms. It was an advanced application of image processing and also is the core basis for our project. Our facial structure was a typical example of a multidimensional structure and needed some recognition from advanced computational analysis

The project aims at designing an intelligent security system for recording the attendance in colleges, firms etc..,. The proposed system makes use of Face Recognition technology for recording the attendance. The system records the attendance along with enter and exit date and time. The Intelligent system proposed makes use of MATLAB software to achieve the task.

We are living in a world where everything is automated and linked online. The internet of things, image processing and machine learning are evolving day by day. Many systems have

been completely changed due to this evolve to achieve more accurate results. The attendance system is a typical example of this transition, starting from the traditional signature on a project sheet to face recognition. In this project we are proposing an automatic attendance system which can be used in every organization to mark the attendance of employees. The main application of Automatic attendance system is seen in teaching institutions, where the attendance of students has to be regularly monitored on daily basis. The method developed provides a secure and effective attendance record. Automatic attendance system uses MATLAB.

The most important thing in the classroom/office is attendance which is directly linked to the academic performance of the students/employees. Recently, some students are husky with better skills during Lectures only when there is massive classroom control .The more efficient the attendance system the mare is participation and learning. In the past we were using techniques like roll numbering calling and signing against a particular roll number. These methods carry a high chance of proxy and are time consuming. We came across are the idea of automating this process through modern day technologies to get a well maintained and disciplined attendance Facial recognition systems along with suitable hardware and software will help meet the goals of this project.

Facial recognition system is derived invocation of image processing deals with the extraction of needy data that can be related to digital image and in technology advancement it plays a unique role. Our core focus will be on receiving digital Images and then making use of programs and algorithms to get useful Information out As the pictorial information is for the image processing work on it and make it set human interpretation. That information from image processing will play a great role and help in various walks of life where it could be implemented. The applications of image processing are vast and can be applied in most scenarios where imaging data could be related to pre-determined algorithms. It was an advanced application of image processing and also is the core basis for our project. Our facial structure was a typical example of a multidimensional structure and needed some recognition from advanced computational analysis.

1.1 Literature Survey

- Authors proposed a model of an automated attendance system. The model focuses on how face recognition incorporated with Radio Frequency Identification (RFID) detect the authorized students and counts as they get in and get out from the classroom. The system keeps the authentic record of every registered student. The system also keeps the data of every student registered for a particular course in the attendance log and provides necessary information according to the need [1].

- In this paper , authors have designed and implemented an attendance system which uses iris biometrics. Initially, the attendees were asked to register their details along with their unique iris template. At the time of attendance, the system automatically took class attendance by capturing the eye image of each attendee, recognizing their iris, and searching for a match in the created database. The prototype was web based [2].

- authors proposed an attendance system based on facial recognition. The algorithms like Viola-Jones and Histogram of Oriented Gradients (HOG) features along with Support Vector Machine (SVM) classifier were used to implement the system. Various real time scenarios such as scaling, illumination, occlusions and pose was considered by the authors. Quantitative analysis was done on the basis of Peak Signal to Noise Ratio (PSNR) values and was implemented in MATLAB GUI [3].

- Authors in [4] researches to get best facial recognition algorithm (Eigenface and Fisherface) provided by the Open CV 2.4.8 by comparing the Receiver Operating Characteristics (ROC) curve and then implemented it in the attendance system. Based on the experiments carried out in this paper, the ROC curve proved that, Eigenface achieves better result than Fisherface. System implemented using Eigenface algorithm achieved an accuracy rate of 70% to 90%.

- In [5], authors proposed a method for student attendance system in classroom using face recognition technique by combining Discrete Wavelet Transforms (DWT) and Discrete Cosine Transform (DCT). These algorithms were used to extract the features of student's face followed by applying Radial Basis Function (RBF) for classifying the facial objects. This system achieved an accuracy rate of 82%.

1.2 Limitations of the existing Work

- The accuracy of the system is not 100%.
- Face detection and loading training data processes are just a little bit slow.
- The instructor and trainingSet manager still have to do some work manually
- It can only detect faces from a limited distance.
- It cannot repeat live video to recognize missed faces.
- People stand in a queue to take their attendance.

1.3 Problem Statement

According to the attendance management system, the accuracy of the data collected is the biggest issue. This is because the attendance might not be recorded personally by the original person, in another word, the attendance of a particular person can be taken by a third party without the realization of the institution which violates the accuracy of the data.

- The second problem is where it is too time consuming. Assuming the time taken for a student to sign his/her attendance on a 3-4 paged name list is approximately 1 minute. In 1 hour, only approximately 60 students can sign their attendance which is obviously inefficient and time consuming.
- The third issue is with the accessibility of those information by the legitimate concerned party. For an example, most of the parents are very concerned to track their child‟s actual whereabouts to ensure their kid really attend the classes in college/school. However there are no ways for the parents to access suchinformation. Therefore, evolution is needed to be done to the system to improve efficiency, data accuracy and provides accessibility to the information for those legitimate party

The concept of face recognition is to give a computer system the ability of finding and recognizing human faces fast and precisely in images or videos. Numerous algorithms and techniques have been developed for improving the performance of face recognition. Recently Deep learning has been highly explored for computer vision applications. Human brain can automatically and instantly detect and recognize multiple faces. But when it comes to computers, it is very difficult to do all the challenging tasks on the level of the human brain.

Face recognition is an integral part of biometrics. In biometrics, basic traits of humans are matched to the existing data. Facial features are extracted and implemented through algorithms, which are efficient and some modifications are done to improve the existing algorithm models. Computers that detect and recognize faces could be applied to a wide variety of practical applications including criminal identification, security systems, identity verification etc. The face recognition system generally involves two stages:

Face Detection – where the input image is searched to find any face, then image processing cleans up the facial image for easier recognition.

Face Recognition – where the detected and processed face is compared to the database of known faces to decide who that person is.

1.4 Objective

- To develop a portable Smart Attendance System which is handy and self-powered.
- Have enough memory space to store the database.
- Able to recognize the face of an individual accurately based on the face database.
- Allow new students or staff to store their faces in the database
- Able to show an indication to the user whether the face- recognition process is successful or not

1.5 Methodology

A. Enrollment Phase

1)ImageAcquisition: For Image acquisition, camera is used to capture an image of the employees working in an organization.

2) Face Detection and Face cropping: For face detection cascading algorithm is used. Further face cropping is done using imcrop (image) command that separates the facial area from the rest of the background image. This faces is stored in the database.

3) Feature Extraction: Feature extraction is done by using linear binary pattern algorithm. Feature extraction is helpful for face detection and recognition.

4) Database: Extracted features of employees or persons will be stored in Database.

B. Routine Attendance after Enrollment Phase

[3] Routine attendance will start after extracting the features of all the images stored in the database. We have followed the same steps as given in enrollment phase i.e. image acquisition, Face detection, and face cropping.

Face recognition is done by using the Local binary pattern by taking 3x3 block a time and comparing central block with surrounding blocks and compare with features stored in the database. If after face recognition face of the person is matched in the database then update the attendance otherwise show the dialog box with invalid person.

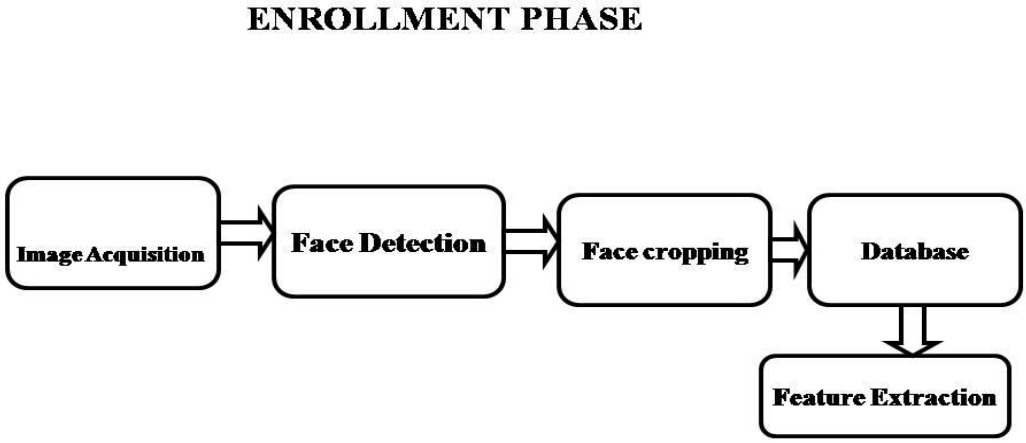


FIG 1.1 Enrollment Phase

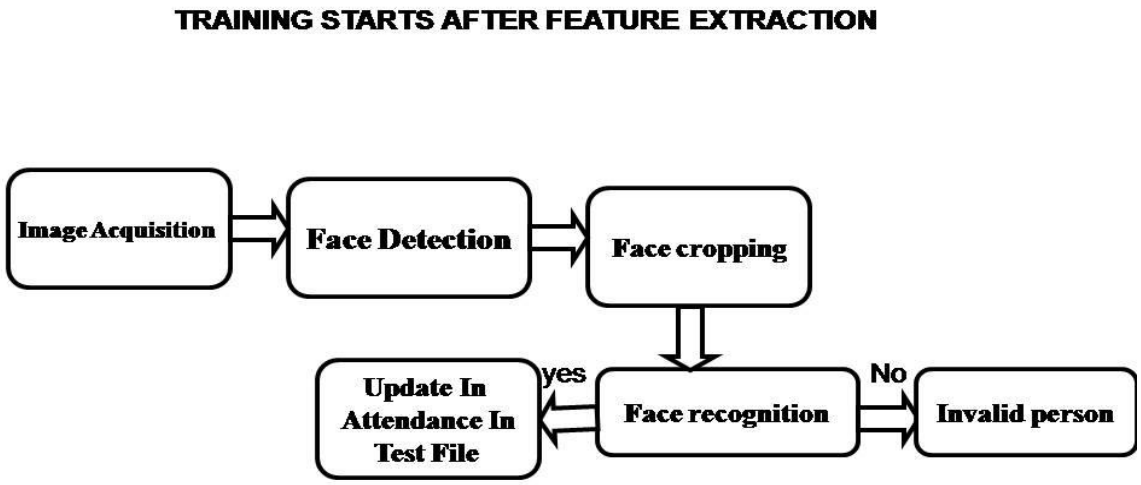


FIG 1.2 Feature Extraction

1.6 Hardware and Software tools used

Software tool

- MATLAB

Hardware tool

- Windows 11 laptop
- Storage
- Camera(Logitech)

Chapter 2

2. Basic theory

During this era of technology and automation we are still using the same old ways of attendance management. The most important thing in the classroom/office is attendance which is directly linked to the academic performance of the students/employees. Recently, some students are husky with better skills during Lectures only when there is massive classroom control (Research Gale, 2018).

The more efficient the attendance system the mare is participation and learning. In the past we were using techniques like roll numbering calling and signing against a particular roll number. These methods carry a high chance of proxy and are time consuming. We came across the idea of automating this process through modern day technologies to get a well maintained and disciplined attendance.

Facial recognition systems along with suitable hardware and software will help meet the goals of this project. Facial recognition system is derived incovalion of image processing Image processing deals with the extraction of needy data that can be related to digital Image and in technology advancement it plays a unique role.

Our core focus will be on receiving digital Images and then making use of programs and algorithms to get useful Information out of it. As the pictorial information is for the image processing work on it and make it esetl human interpretation. That information from image processing will play a great role and help in various walks of life where it could be implemented.

The applications of image processing are vast and can be applied in most scenarios where imaging data could be related to pre-determined algorithms. It was an advanced application of image processing and also is the core basis for our project. Our facial structure was a typical example of a multidimensional structure and needed some recognition from advanced computational analysis.

During this era of technology and automation we are still using the same old ways of attendance management. The most important thing in the classroom/office is attendance which is directly linked to the academic performance of the students/employees.

Recently, some students are husky with better skills during Lectures only when there is massive classroom control .The more efficient the attendance system the mare is participation and learning. In the past we were using techniques like roll numbering calling and signing against a particular roll number. These methods carry a high chance of proxy and are time consuming. We came across are the idea of automating this process through modern day technologies to get a well maintained and disciplined attendance, Facial recognition systems along with suitable hardware and software will hellp meet the goals ofthis project.

Chapter 3

3.TOOL DESCRIPTION

MATLAB

The name MATLAB stands for matrix laboratory. MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran.

MATLAB can be used in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. For a million engineers and scientists in industry and academia, MATLAB is the language of technical computing.

The MATLAB System

The MATLAB system consists of five main parts:

- The MATLAB language.

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

- The MATLAB working environment.

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

- Handle Graphics.

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and

presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

- The MATLAB mathematical function library.

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

- The MATLAB Application Program Interface (API).

This is a library that allows you to write C and Fortran programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

MATLAB is a high-level language: MATLAB Supports Object oriented programming. It also supports different types of programming constructs like Control flow statements (IF-ELSE, FOR, WHILE). MATLAB also supports structures like in C programming, Functional programming (writing functions to contain commonly used code and later calling them). It also contains Input / Output statements like disp() and input().
Interactive graphics: MATLAB has inbuilt graphics to enhance user experience. We can actually visualize whatever data is there in forms of plots and figures. It also supports processing of image and displaying them in 2D or 3D formats. We can visualize and manipulate our data across any of the three dimensions (1D, 2D, and 3D). We can plot the functions and customize them also according to our needs like changing bullet points, line color and displaying/not displaying grid.
A large library of Mathematical functions: MATLAB has a huge inbuilt library of functions required for mathematical analysis of any data. It has common math functions like sqrt. factorial etc. It has functions required for statistical analysis like median, mode and std (to find standard deviation), and much more. MATLAB also has functions for signal processing like filter, butter(Butterworth filter design) audio read, Conv, xcorr, fft, fftshift etc. It also supports image processing and some common functions required for image processing in MATLAB are rgb2gray, rgb2hsv, adaptthresh etc.
Data access and processing: MATLAB allows accessing of data from external sources like image files (.jpg, .PNG), audio files (.mp), and real-time data from JDBC/ ODBC. We can

easily read data from external sources using the inbuilt MATLAB functions like audioread for reading audio files and imread for reading external images.

Interactive environment: MATLAB offers interactive environment by providing a GUI (Graphical user interface) and different types of tools like signal analyses and tuners. MATLAB also has tools for debugging and the development of any software. Importing and exporting files becomes easy in MATLAB through the GUI. We can view the workspace data as we progress in the development of our software and modify it according to our needs.

MATLAB can interface with different languages: We can write a set of codes (libraries) in languages like PERL and JAVA, and we can call those libraries from within the MATLAB itself. MATLAB also supports ActiveX and .NET libraries.

MATLAB and Simulink : MATLAB has an inbuilt feature of Simulink wherein we can model the control systems and see their real-time behavior. We can design any system either using code or building blocks and see their real-time working through various inbuilt tools. It has lucid examples of basic control systems and their working.

MATLAB's Application programming interface (API): MATLAB consists of an extensive API. Through this API, we can link our C/C++ programs directly to MATLAB. Some options available in MATLAB API are calling MATLAB programs, read and write M-files, and using MATLAB as an interface to run applications. MATLAB can be used both as a computation and analysis tool.

Machine Learning, Deep Learning, and Computer vision: The most demanding technologies like Machine learning, Deep learning, and Computer vision can be done in MATLAB. We can create and interconnect layers of a deep neural network, We can build custom training loops and training layers with automatic differentiation. For machine learning, we can use the DBSCAN algorithm to discover clusters and noise in DATA. For computer vision, we can do object tracking, object recognition, gesture recognition, and processing 3D point clouds.

Computational Biology toolbox: This toolbox provides a great way for biologists and researchers to create and analyze new algorithms and patterns for development in biological and biochemical domains. We can build biological models and analyze them using this toolbox. Moreover, for students, this toolbox can be very much educational if they want to explore the biological domain.

Chapter 4

4. Implementation

4.1 Hardware Design and Implementation

4.1.1 Start with MATLAB

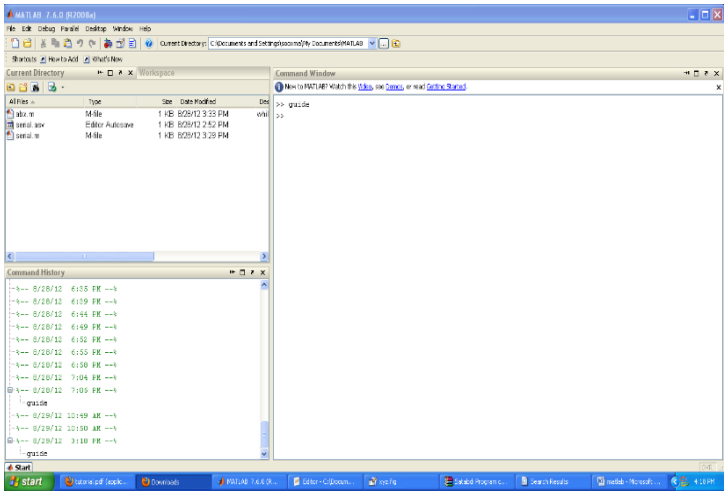


FIG 4.1 Start with MATLAB

This editor window will be opened.

For any help regarding MATLAB just type *doc help* on MATLAB command window. And this editor window will be opened. Now you have to type the name of the particular field you are searching for. And you will get all the information regarding that figure

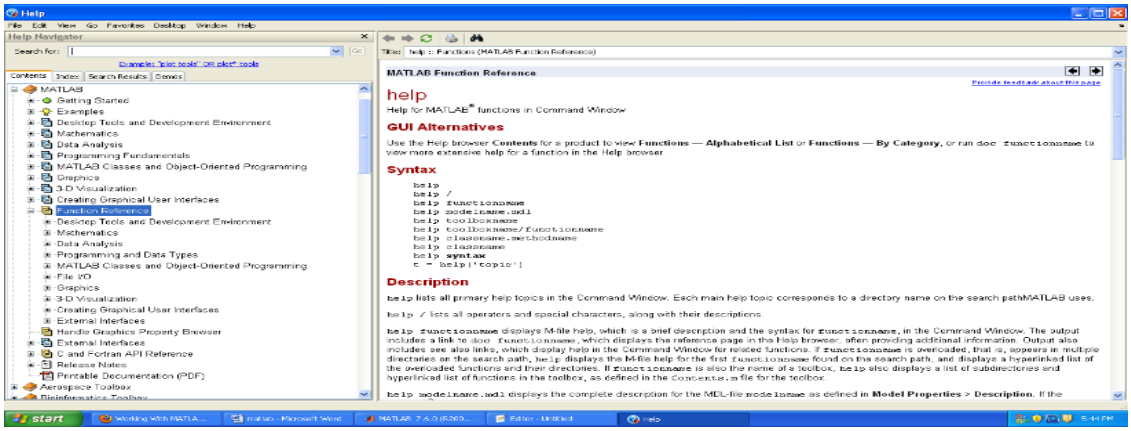


FIG 4.2 Editor window

- Creating Variables: x=5.71;
- Creating String: Mystring= ’Hello, world’
- Mathematical use: *Mathematical symbols*

‘+’=Addition, ‘-‘=Subtraction, ‘*’=Multiplication,‘/’=Division,^ = Power operator ,'= Transpose

Mathematical Functions: sqrt(x)-square root,sin(x)-sine function.cos(x),tan(x) are also there.exp(x)-exponential, log(x)-natural log, 10log(x)-common log, abs(x)-absolutevalue,mod(x)-modulas,factorial(x)-factorial function,angle(x+iy)- angle of a complex number, in radians,min(x)-minimum element of an array. max(x) is also there,floor(x)-round down. ceil(x) and round(x) are also there. Besselj(x)-besses function

MATLAB also has built in constants like pi (π) and i(imaginary number).

- Creating Matrix:

The best data element in MATLAB is Matrix

A= [1 2 3 ; 4 5 6; 7 8 9];

Useful matrices: zeros-creates a matrix of zeros, ones- creates a matrix of ones, rand- creates a matrix of random numbers, eye-creates an identity matrix.

4.1.2 FILE EXTENSIONS IN MATLAB:

- Native

.fig-MATLAB Figure

.m -MATLAB function, script, or class

.mat -MATLAB binary file for storing variables

.mex-MATLAB executable (platform specific, e.g. ".mexmac" for the Mac, ".mexglx" for Linux, etc.)

.p -MATLAB content-obscured .m file (result e())

- Third-party

.jkt -GPU Cache file generated by Jacket for MATLAB (AccelerEyes)

.mum -MATLAB CAPE-OPEN Unit Operation Model File (AmsterCHEM)

4.1.3 M FILES:

MATLAB can also be used as a programming language. To program in MATLAB on text file should be created in MATLAB containing MATLAB commands as you would type them interactively in MATLAB window. The may have any legal unix name and should be ended with extension .m. This file may be placed in root directory or in a directory named /MATLAB.

There are two types of m-files in MATLAB. One is called script. It contains only the list of MATLAB commands without any header. And the other type is function, which contains a header like this:

Function y=f(cmdd)

This function may pass arguments or results depending upon the codes.

To create new M-File follow this commands.

File-New-M-File

And this editor window will be opened.

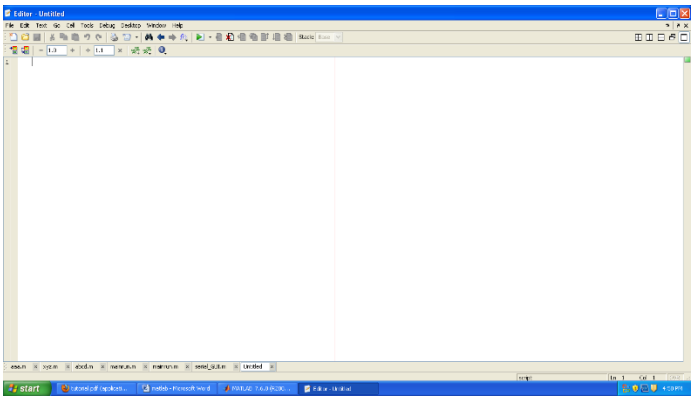


FIG 4.3 M FILE WINDOW

In this file all the commands should be written. Then save the program and run it by pressing F5 button on your keyboard or by pressing this  button on your window.

Basic Syntax

```
if        - Conditionally execute statements.
else      - IF statement condition.
elseif    - IF statement condition.
end       - Terminate scope of FOR, WHILE, SWITCH, TRY and IF statements.
for       - Repeat statements a specific number of times.
while     - Repeat statements an indefinite number of times.
break     - Terminate execution of WHILE or FOR loop.
continue  - Pass control to the next iteration of FOR or WHILE loop.
function  - Add new function.
return    - Return to invoking function.
```

```
error     - Display error message and abort function.
disp      - Display an array.
feval     - Execute function specified by string.
```

4.1.4 PLOTTING USING MATLAB

MATLAB provides very good 2D and 3D plots and 3D visualization functions.

List of commands used for plotting

Plot, xlabel, ylabel, title, grid, axis, stem, subplot

- TWO DIMENTIONAL PLOT

To draw 2D plot **plot** command is used. The syntax is **plot(x,y)**

x is a vector containing x-coordinates of the plot and

y is a vector containing x-coordinates of the plot.

To plot a curve of equation x=y^2 where x varies from 0 to 10. The commands will be

x=0 :1 : 10;

y=x..^2;

plot(x,y);

xlabel('x axis'); to give a name for x coordinate

ylabel(‘y axis’); to give a name for x coordinate

title(‘x Vs y plotting’); to give a name for x coordinate

grid on to get grid on the plotting

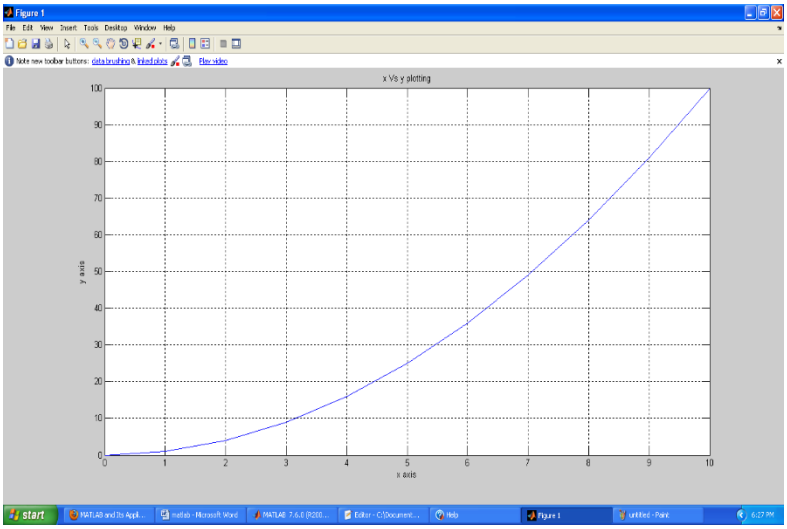


FIG 4.4 PLOTTING WINDOW

We can use color and different type of marker style and line style in MATLAB as written in the box bellow

Colour Style Options		Marker Style Options		Line Style Options	
y	yellow	.	point	-	solid
m	magenta	o	circle	:	dotted
c	cyan	x	x-mark	-.	dashdot
r	red	+	plus	--	dashed
g	green	*	star		
b	blue	s	square		
w	white	d	diamond		
k	black	v	triangle(down)		
		^	triangle (up)		
		<	triangle(left)		
		>	triangle(right)		
		p	pentagram		
		h	hexagram		

For example, plot(x,y,'--') uses a dashed line, plot(x,y,'*') uses * at all the points defined in t and y without connecting the points, and plot(x,y,'g') uses a solid green line. The options can also be used together, for example, plot(x,y,'g:') plots a dotted green line.

To plot two or more graphs on the same set of axes, use the command plot(x1,y1,x2,y2), which plots x1 versus t1 and x2 versus t2.

The problem that you will encounter most often when plotting functions is that MATLAB will scale the axes in a way that is different than you want them to appear. You can easily override the auto scaling of the axes by using the axis command after the plotting command:

axis([xmin xmax ymin ymax]);

where xmin, xmax, ymin, and ymax are numbers corresponding to the limits you desire for the axes. To return to the automatic scaling, simply type axis.

For discrete-time signals, use the command stem which plots each point with a small open circle and a straight line. To plot y[x] versus x, type stem(x,y). To obtain filled circles, as the plots appear in the textbook, use the command stem(x,y,'filled').

To plot more than one graph on the screen, use the command subplot(mnp) which partitions the screen into an mxn grid where p determines the position of the particular graph counting the upper left corner as p=1. For example,

```
subplot(211),semilogx(w,magdb);

subplot(212),semilogx(w,phase);
```

plots the bode plot with the log-magnitude plot on top and the phase plot below. Titles and labels can be inserted immediately after the appropriate semilogx command or plot command. To return to a full screen plot, type subplot(111).

Enabling 3-D Rotation

You can easily rotate graphs to any orientation with the mouse. Rotation involves the reorientation of the axes and all the graphics objects it contains. Therefore none of the data defining the graphics objects is affected by rotation; instead the orientation of the *x*-, *y*-, and *z*-axes changes with respect to the viewer.

There are three ways to enable Rotate 3D mode:

- Select Rotate 3D from the Tools menu.
- Click the Rotate 3D icon in the figure toolbar  .
- Execute the *rotate3d* command.

Once the mode is enabled, you press and hold the mouse button while moving the cursor to rotate the graph.

Selecting Predefined Views

When Rotate 3D mode is enabled, you can control various rotation options from the right-click context menu.

You can rotate to predefined views on the right-click context menu:

- Reset to Original View — Reset to the default view (azimuth -37.5°, elevation 30°).
- Go to X-Y View — View graph along the *z*-axis (azimuth 0°, elevation 90°).
- Go to X-Z View — View graph along the *y*-axis (azimuth 0°, elevation 0°).
- Go to Y-Z View — View graph along the *x*-axis (azimuth 90°, elevation 0°).

Rotation Style for Complex Graphs

You can select from two rotation styles on the right-click context menu's Rotation Options submenu:

- Plot Box Rotate — Display only the axes bounding box for faster rotation of complex objects. Use this option if the default Continuous Rotate style is unacceptably slow.
- Continuous Rotate — Display all graphics during rotation.

Axes Behavior During Rotation

You can select two types of behavior with respect to the aspect ratio of axes during rotation:

- Stretch-to-Fill Axes – Default axes behavior is optimized for 2-D plots. Graphs fit the rectangular shape of the figure.

- Fixed Aspect Ratio Axes – Maintains a fixed shape of objects in the axes as they are rotated. Use this setting when rotating 3-D plots.

Undo/Redo — Eliminating Mistakes

The figure Edit menu contains two items that enable you to undo any zoom, pan, or rotate operation.

Undo — Remove the effect of the last operation.

Redo — Perform again the last operation that you removed by selecting Undo.

4.1.5 SERIAL COMMUNICATION USING MATLAB:

Serial communication is the most common low-level protocol for communicating between two or more devices. Normally, one device is a computer, while the other device can be a modem, a printer, another computer, microcontroller or a scientific instrument such as an oscilloscope or a function generator.

As the name suggests, the serial port sends and receives bytes of information in a serial fashion — one bit at a time. These bytes are transmitted using either a binary (numerical) format or a text format.

The serial port interface for connecting two devices is specified by the TIA/EIA-232C standard published by the Telecommunications Industry Association.

The original serial port interface standard was given by RS-232, which stands for Recommended Standard number 232. The term *RS-232* is still in popular use, and is used in this guide when referring to a serial communication port that follows the TIA/EIA-232 standard. RS-232 defines these serial port characteristics:

- The maximum bit transfer rate and cable length
- The names, electrical characteristics, and functions of signals
- The mechanical connections and pin assignments

Primary communication is accomplished using three pins: the Transmit Data pin, the Receive Data pin, and the Ground pin. Other pins are available for data flow control, but are not required.

Other standards such as RS-485 define additional functionality such as higher bit transfer rates, longer cable lengths, and connections to as many as 256 devices.

The MATLAB serial port interface provides direct access to peripheral devices such as modems, printers, and scientific instruments that you connect to your computer's serial port. This interface is established through a serial port object. The serial port object supports functions and properties that allow you to

- Configure serial port communications
- Use serial port control pins
- Write and read data
- Use events and callbacks
- Record information to disk

Instrument Control Toolbox™ software provides additional serial port functionality. In addition to command-line access, this toolbox has a graphical tool called the Test & Measurement Tool, which allows you to communicate with, configure, and transfer data with your serial device without writing code. The Test & Measurement Tool generates MATLAB code for your serial device that you can later reuse to communicate with your device or to develop GUI-based applications. The toolbox includes additional serial I/O utility functions that facilitate object creation and configuration, instrument communication, and so on. With the toolbox you can communicate with GPIB- or VISA-compatible instruments.

Using MATLAB, we can select COMPORT, BAUDRATE, START BIT, STOP BIT, PARITY BIT and DATA BIT for serial communication.

- *Create a serial port object:*

Obj=serial(‘port’,’propertyname’,propertyvalue)

Example: s = serial(‘COM1’,’BaudRate’,9600);

To specify properties during object creation

s = serial('COM2','BaudRate',9600,'DataBits',8);

Obj = serial('*port*') creates a serial port object associated with the serial port specified by *port*. If *port* does not exist, or if it is in use, you will not be able to connect the serial port object to the device.

Port object name will depend upon the platform that the serial port is on. instrhwinfo ('serial') provides a list of available serial ports. This list is an example of serial constructors on different platforms:

- *Open serial port:*

fopen(‘obj’)

Example: fopen(s);

- *Write in the serial port*

fwriets(‘obj’,’variable name’); or fprintf(‘obj’,’variable name’);

Example:fwrites(s,str); or fprintf(s,str);

- *Read from the serial port:*

‘variable name’=fscanf(‘obj’); or

‘variable name’=freads(‘obj’); or to get binary data

‘variable name’=fgetc(‘obj’); or to get char data

‘variable name’=fgets(‘obj’); or to get string data

Example

x=fscanf(s);

x=freads(s);

x=fgetc(s);

x=fgets(s);

- Close the serial port:

fclose('obj');

Example:

fclose(s);

- Remove the serial port from memory:

delete(obj)

Example:

delete(s);

When you delete obj, it becomes an *invalid* object. Because you cannot connect an invalid serial port object to the device, you should remove it from the workspace with the clear command. If multiple references to obj exist in the workspace, then deleting one reference invalidates the remaining references.

If obj is connected to the device, it has a Status property value of open. If you issue delete while obj is connected, then the connection is automatically broken.

- Remove serial port object from MATLAB workspace:

clear obj

Example:

clear s

If obj is connected to the device and it is cleared from the workspace, then obj remains connected to the device. You can restore obj to the workspace with the instrfind function. A serial port object connected to the device has a Status property value of open.

- Read serial port objects from memory to MATLAB workspace:

out=instrfind
out=instrfind('PropertyName',PropertyValue,...)
out=instrfind(S)
out = instrfind(obj,'PropertyName',PropertyValue,...)

Example:

Suppose you create the following two serial port objects on a Windows platform.

```
s1 = serial('COM1');
s2 = serial('COM2');
set(s2,'BaudRate',4800)
fopen([s1 s2])
```

You can use instrfind to return serial port objects based on property values.

```
out1 = instrfind('Port','COM1');
out2 = instrfind({'Port','BaudRate'},{'COM2',4800});
```

You can also use instrfind to return cleared serial port objects to the MATLAB workspace.

```
clear s1 s2
newobjs = instrfind

  Instrument Object Array
  Index:  Type:       Status:    Name:
  1       serial      open       Serial-COM1
  2       serial      open       Serial-COM2
```

To close both s1 and s2

fclose(newobjs)

So using MATLAB we can get data serially from any other norms like another PC or Microcontroller and store them in an array and finally we can draw a real time plot using those values.

4.1.6 MATLAB COMMAND LIST:

Operators and Special Characters	
+	Plus; addition operator.
−	Minus; subtraction operator.
*	Scalar and matrix multiplication operator.
.*	Array multiplication operator.
^	Scalar and matrix exponentiation operator.
.^	Array exponentiation operator.
\	Left-division operator.
/	Right-division operator.
.\	Array left-division operator.
./	Array right-division operator.
:	Colon; generates regularly spaced elements and represents an entire row or column.
( )	Parentheses; encloses function arguments and array indices; overrides precedence.
[ ]	Brackets; enclosures array elements.
.	Decimal point.
...	Ellipsis; line-continuation operator.
,	Comma; separates statements and elements in a row.
;	Semicolon; separates columns and suppresses display.
%	Percent sign; designates a comment and specifies formatting.
_	Quote sign and transpose operator.
._	Nonconjugated transpose operator.
=	Assignment (replacement) operator.

Commands for Managing a Session	
`clc`	Clears Command window.
`clear`	Removes variables from memory.
`exist`	Checks for existence of file or variable.
`global`	Declares variables to be global.
`help`	Searches for a help topic.
`lookfor`	Searches help entries for a keyword.
`quit`	Stops MATLAB.
`who`	Lists current variables.
`whos`	Lists current variables (long display).

Special Variables and Constants	
`ans`	Most recent answer.
`eps`	Accuracy of floating-point precision.
`i,j`	The imaginary unit $\sqrt{-1}$.
`Inf`	Infinity.
`NaN`	Undefined numerical result (not a number).
`pi`	The number π .

System and File Commands	
`cd`	Changes current directory.
`date`	Displays current date.
`delete`	Deletes a file.
`diary`	Switches on/off diary file recording.
`dir`	Lists all files in current directory.
`load`	Loads workspace variables from a file.
`path`	Displays search path.
`pwd`	Displays current directory.
`save`	Saves workspace variables in a file.
`type`	Displays contents of a file.
`what`	Lists all MATLAB files in the current directory.
`wklread`	Reads .wk1 spreadsheet file.

Input/Output and Formatting Commands

Input/Output Commands	
`disp`	Displays contents of an array or string.
`fscanf`	Read formatted data from a file.
`format`	Controls screen-display format.
`fprintf`	Performs formatted writes to screen or file.
`input`	Displays prompts and waits for input.
`;`	Suppresses screen printing.

Format Codes for `fprintf` and `fscanf`	
`%s`	Format as a string.
`%d`	Format as an integer.
`%f`	Format as a floating point value.
`%e`	Format as a floating point value in scientific notation.
`%g`	Format in the most compact form: %f or %e.
`\n`	Insert a new line in the output string.
`\t`	Insert a tab in the output string.

Numeric Display Formats	
`format short`	Four decimal digits (default).
`format long`	16 decimal digits.
`format short e`	Five digits plus exponent.
`format long e`	16 digits plus exponents.
`format bank`	Two decimal digits.
`format +`	Positive, negative, or zero.
`format rat`	Rational approximation.
`format compact`	Suppresses some line feeds.
`format loose`	Resets to less compact display mode.

Vector, Matrix and Array Commands

Array Commands	
cat	Concatenates arrays.
find	Finds indices of nonzero elements.
length	Computers number of elements.
linspace	Creates regularly spaced vector.
logspace	Creates logarithmically spaced vector.
max	Returns largest element.
min	Returns smallest element.
prod	Product of each column.
reshape	Change size
size	Computes array size.
sort	Sorts each column.
sum	Sums each column.

Special Matrices	
eye	Creates an identity matrix.
ones	Creates an array of ones.
zeros	Creates an array of zeros.

Matrix Arithmetic	
cross	Computes cross products.
dot	Computes dot products.

Matrix Commands for Solving Linear Equations	
det	Computes determinant of an array.
inv	Computes inverse of a matrix.
pinv	Computes pseudoinverse of a matrix.
rank	Computes rank of a matrix.
rref	Computes reduced row echelon form.

Cell Array Functions

`cell`	Creates cell array.
`celldisp`	Displays cell array.
`cellplot`	Displays graphical representation of cell array.
`num2cell`	Converts numeric array to cell array.
`deal`	Matches input and output lists.
`iscell`	Identifies cell array.

Structure Functions

`fieldnames`	Returns field names in a structure array.
`getfield`	Returns field contents of a structure array.
`isfield`	Identifies a structure array field.
`isstruct`	Identifies a structure array.
`rmfield`	Removes a field from a structure array.
`setfield`	Sets contents of field.
`struct`	Creates structure array.

Plotting Commands

Basic xy Plotting Commands	
`axis`	Sets axis limits.
`fplot`	Intelligent plotting of functions.
`grid`	Displays gridlines.
`plot`	Generates xy plot.
`print`	Prints plot or saves plot to a file
`title`	Puts text at top of plot.
`xlabel`	Adds text label to x-axis.
`ylabel`	Adds text label to y-axis.

Plot Enhancement Commands	
axes	Creates axes objects.
close	Closes the current plot.
close all	Closes all plots.
figure	Opens a new figure window.
gtext	Enables label placement by mouse.
hold	Freezes current plot.
legend	Legend placement by mouse.
refresh	Redraws current figure window.
set	Specifies properties of objects such as axes.
subplot	Creates plots in subwindows.
text	Places string in figure.

Specialized Plot Commands	
bar	Creates bar chart.
loglog	Creates log-log plot.
polar	Creates polar plot.
semilogx	Creates semilog plot (logarithmic abscissa).
semilogy	Creates semilog plot (logarithmic ordinate).
stairs	Creates stairs pot.
stem	Creates stem plot.

Three-Dimensional Plotting Commands	
contour	Creates contour plot.
mesh	Creates three-dimensional mesh surface plot.
meshc	Same as mesh with contour plot underneath.
meshz	Same as mesh with vertical lines underneath.
plot3	Creates three-dimensional plots from lines and points.
surf	Creates shaded three-dimensional mesh surface plot.
surfc	Same as surf with contour plot underneath.
meshgrid	Creates rectangular grid.
waterfall	Same as mesh with mesh lines in one direction.
zlabel	Adds text label to *z*-axis.

Histogram Functions	
bar	Creates a bar chart.
hist	Aggregates the data into equally spaced bins.
histc	Aggregates the data into unequally spaced bins.

Programming

Logical and Relational Operators	
==	Relational operator: equal to.
~=	Relational operator: not equal to.
<	Relational operator: less than.
<=	Relational operator: less than or equal to.
>	Relational operator: greater than.
>=	Relational operator: greater than or equal to.
&	Logical operator: AND.
|	Logical operator: OR.
~	Logical operator: NOT.
xor	Logical operator: EXCLUSIVE OR.

Program Flow Control	
break	Terminates execution of a loop.
case	Provides alternate execution paths within switch structure.
else	Delineates alternate block of statements.
elseif	Conditionally executes statements.
end	Terminates for, while, and if statements.
error	Display error messages.
for	Repeats statements a specific number of times
if	Executes statements conditionally.
otherwise	Default part of switch statement.
return	Return to the invoking function.
switch	Directs program execution by comparing point with case expressions.
warning	Display a warning message.
while	Repeats statements an indefinite number of times.

Logical Functions	
any	True if any elements are nonzero.
all	True if all elements are nonzero.
find	Finds indices of nonzero elements.
finite	True if elements are finite.
isnan	True if elements are undefined.
isinf	True if elements are infinite.
isempty	True if matrix is empty.
isreal	True if all elements are real.

M-Files	
`eval`	Interpret strings containing Matlab expressions.
`feval`	Function evaluation.
`function`	Creates a user-defined function M-file.
`global`	Define global variables.
`nargin`	Number of function input arguments.
`nargout`	Number of function output arguments.
`script`	Script M-files

Timing	
`cputime`	CPU time in seconds.
`clock`	Current date and time as date vector.
`tic, toc`	Start, stop a stopwatch timer.

Mathematical Functions

Exponential and Logarithmic Functions	
`exp(x)`	Exponential; e^x.
`log(x)`	Natural logarithm; ln(x).
`log10(x)`	Common (base 10) logarithm; log(x)= $\log_{10}$(x).
`sqrt(x)`	Square root; $\sqrt{x}$.

Trigonometric Functions	
`acos(x)`	Inverse cosine; arcos x = $\cos^{-1}$ (x).
`acot(x)`	Inverse cotangent; arccot x = $\cot^{-1}$(x).
`acsc(x)`	Inverse cosecant; arcs x = $\csc^{-1}$ (x).
`asec(x)`	Inverse secant; arcsec x = $\sec^{-1}$ (x).
`asin(x)`	Inverse sine; arcsin x = $\sin^{-1}$ (x).
`atan(x)`	Inverse tangent; arctan x = $\tan^{-1}$ (x).
`atan2(y,x)`	Four-quadrant inverse tangent.
`cos(x)`	Cosine; cos(x).
`cot(x)`	Cotangent; cot(x).
`csc(x)`	Cosecant; csc(x).
`sec(x)`	Secant; sec(x).
`sin(x)`	Sine; sin(x).
`tan(x)`	Tangent; tan(x).

Hyperbolic Functions	
`acosh(x)`	Inverse hyperbolic cosine; $\cosh^{-1}(x)$.
`acoth(x)`	Inverse hyperbolic cotangent; $\coth^{-1}(x)$.
`acsch(x)`	Inverse hyperbolic cosecant; $\mathrm{csch}^{-1}(x)$.
`asech(x)`	Inverse hyperbolic secant; $\mathrm{sech}^{-1}(x)$.
`asinh(x)`	Inverse hyperbolic sine; $\sinh^{-1}(x)$.
`atanh(x)`	Inverse hyperbolic tangent; $\tanh^{-1}(x)$.
`cosh(x)`	Hyperbolic cosine; cosh(x).
`coth(x)`	Hyperbolic cotangent; cosh(x)/sinh(x).
`csch(x)`	Hyperbolic cosecant; 1/sinh(x).
`sech(x)`	Hyperbolic secant; 1/cosh(x).
`sinh(x)`	Hyperbolic sine; sinh(x).
`tanh(x)`	Hyperbolic tangent; sinh(x)/cosh(x).

Complex Functions	
`abs(x)`	Absolute value; |x|.
`angle(x)`	Angle of a complex number x.
`conj(x)`	Complex conjugate of x.
`imag(x)`	Imaginary part of a complex number x.
`real(x)`	Real part of a complex number x.

Statistical Functions	
`erf(x)`	Computes the error function *erf* (x).
`mean`	Calculates the average.
`median`	Calculates the median.
`std`	Calculates the standard deviation.

Random Number Functions	
`rand`	Generates uniformly distributed random numbers between 0 and 1.
`randn`	Generates normally distributed random numbers.

Numeric Functions	
ceil	Rounds to the nearest integer toward ∞.
fix	Rounds to the nearest integer toward zero.
floor	Rounds to the nearest integer toward - ∞.
round	Rounds towards the nearest integer.
sign	Signum function.

String Functions	
findstr	Finds occurrences of a string.
strcmp	Compares strings.
char	Creates character string array

Numerical Methods

Polynomial and Regression Functions	
conv	Computes product of two polynomials
deconv	Computes ratio of polynomials.
eig	Computes the eigenvalues of a matrix.
poly	Computes polynomial from roots.
polyfit	Fits a polynomial to data.
polyval	Evaluates polynomial and generates error estimates.
roots	Computes polynomial roots.

Interpolation Functions	
interp1	Linear and cubic-spline interpolations of a function of one variable.
interp2	Linear interpolation of a function of two variables.
spline	Cubic-spline interpolation.
unmkpp	Computes the coefficients of cubic-spine polynomials.

Root Finding and Minimization	
fmin	Finds minimum of single-variable function.
fmins	Finds minimum of multivariable function.
fzero	Finds zero of single-variable function.

Numerical Integration Functions	
quad	Numerical integration with adaptive Simpson's rule.
quadl	Numerical integration with adaptive Lobatto quadrature.
trapz	Numerical integration with the trapezoidal rule.

Numerical Differentiation Functions	
diff(x)	Computes the difference between adjacent elements in the vector x.
polyder	Differentiates a polynomial, a polynomial product, or a polynomial quotient.

ODE Solvers	
ode23	Nonstiff, low-order solver.
ode45	Nonstiff, medium-order solver.
ode113	Nonstiff, variable-order solver.
ode23s	Stiff, low-order.
ode23t	Moderately stiff, trapezoidal rule solver.
ode23b	Stiff, low-order solver.
ode15s	Stiff, variable-order solver.
odeset	Creates integrator options structure for ODE solvers.

Predefined Input Functions	
gensig	Generates a periodic sine, square, or pulse input.
sawtooth	Generates a periodic sawtooth input.
square	Generates a square wave input.
stepfun	Generates a step function input.

Symbolic Math Toolbox

Functions for Creating and Evaluating Symbolic Expressions	
class	Returns the class of an expression.
digits	Sets the number of decimal digits used to do variable precision arithmetic.
double	Converts an expression to numeric form.
ezplot	Generates a plot of a symbolic expression.
findsym	Finds the symbolic variables in a symbolic expression.
numden	Returns the numerator and denominator of an expression.
sym	Creates a symbolic variable.
syms	Creates one or more symbolic variables.
vpa	Sets the number of digits used to evaluate expressions.

Functions for Manipulating Symbolic Expressions	
collect	Collects coefficients of like powers in an expression.
expand	Expands an expression by carrying out jpowers.
factor	Factors an expression.
poly2sym	Converts a polynomial coefficient vector to a symbolic polynomial.
pretty	Displays an expression in a form that resembles typeset mathematics.
simple	Searches for the shortest form of an expression.
simplify	Simplifies an expression using Maple's simplification rules.
subs	Substitutes variables or expressions.
sym2poly	Converts an expression to a polynomial coefficient vector.

Symbolic Calculus Functions	
diff	Returns the derivative of an expression.
Dirac	Dirac delta function (unit impulse).
Heaviside	Heaviside function (unit step).
int	Returns the integral of an expression.
limit	Returns the limit of an expression.
symsum	Returns the symbolic summation of an expression.
taylor	Returns the Taylor series of a function.

Symbolic Solution of Algebraic and Transcendental Equations	
solve	Solves symbolic equations.

Symbolic Solution of Differential Equations	
dsolve	Returns a symbolic solution of a differential equation or set of equations.

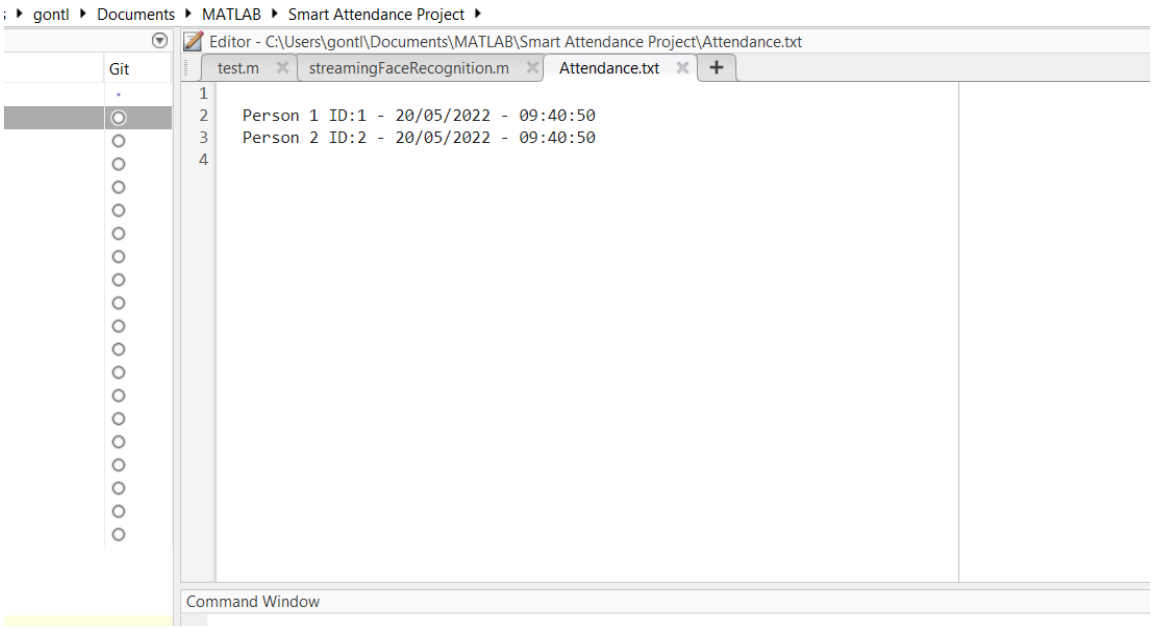
Laplace Transform Functions	
ilaplace	Returns the inverse Laplace transform.
laplace	Returns the Laplace transform.

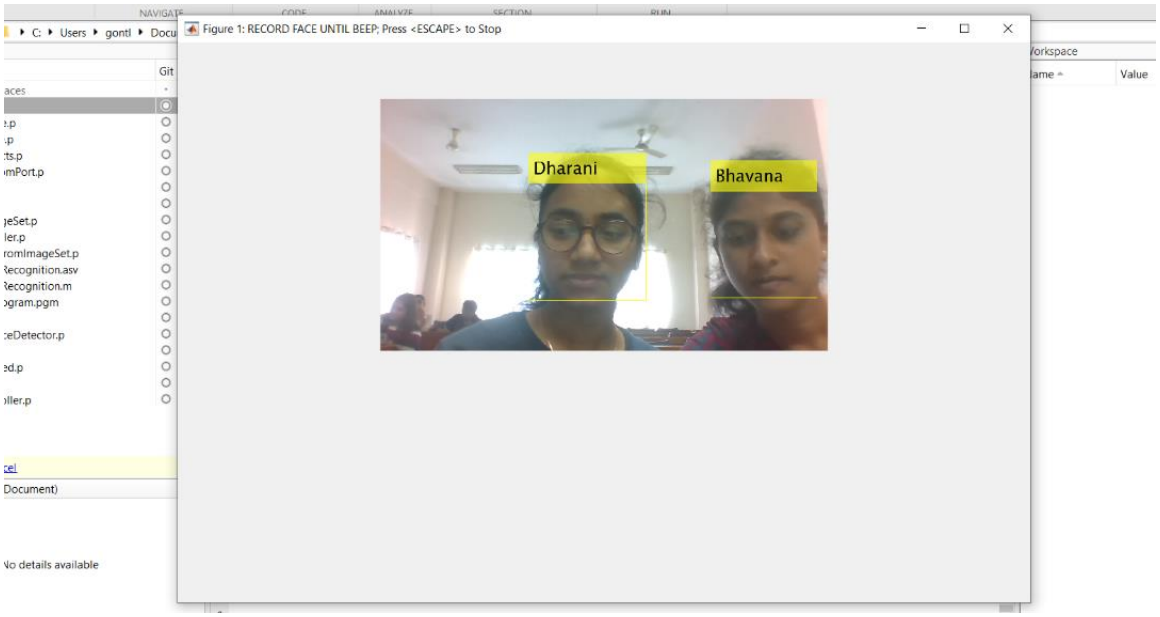
Symbolic Linear Algebra Functions	
det	Returns the determinant of a matrix.
eig	Returns the eigenvalues (characteristic roots) of a matrix.
inv	Returns the inverse of a matrix.
poly	Returns the characteristic polynomial of a matrix.

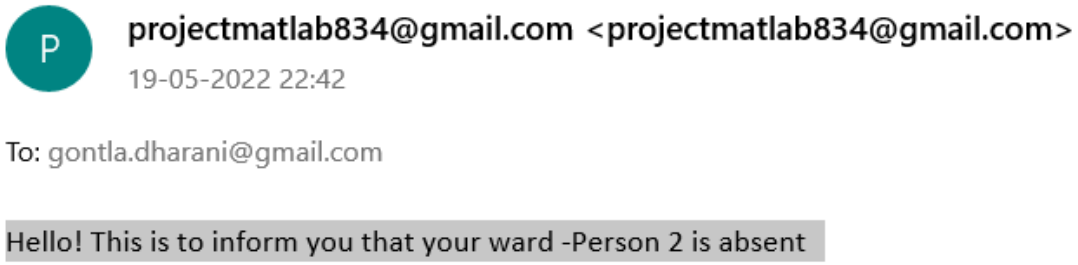
CHAPTER 5:

5. Result and discussion

The project "**SMART ATTENDANCE SYSTEM USING FACE RECOGNITION**" was designed an intelligent security system for recording the attendance in colleges, firms etc..,. The proposed system makes use of Face Recognition technology for recording the attendanceusing MATLAB software.







The topic chosen by the group was a digital system on facial detection and facial recognition. At the start of the project, the group's objectives were to represent a digital system in a simple yet effective way, to successfully simulate a facial detection and recognition program, to learn more about the functions that MATLAB has to offer, and to increase their knowledge on the various applications that digital communication has. The eyes of the group were opened to the various possible topics that they could tackle and the effects that each topic has in the world that we live in today. There search of the group led to the realization about how the modernization of technology brings even more possibilities and ideas for the future of digital communications

Conclusion and future scope:

With the advent of fast and cheap machines, digital image processing has become a very highly demanded field of study and practice. It provides solutions to various real-life applications in an economical way.

Future Scope: Almost all academic institutions require attendance records of students and maintaining attendance manually can be hectic as well as time consuming.

Hence maintaining attendance automatically with the help of face recognition will be very helpful and less prone to errors as compared to manual process.

REFRENCES

[1] Muhammad Fuzail, Hafiz Muhammad Fahad Nouman, Muhammad Omer Mushtaq, Binish Raza, Awais Tayyab, and Muhammad Waqas Talib, "Face Detection System for Attendance of Class' Students", International Journal Of Multidisciplinary Sciences And Engineering, vol. 5, no. 4, April 2014.

[2]Adrian Rhesa Septian Siswanto, Anto Satriyo Nugroho, MaulahikmahGalinium,"Implementation of Face Recognition Algorithm for Biometrics Based Time Attendance System", IEEE Conference.

[3] Aziza Ahmedi , 2Dr Suvarna Nandyal, "An Automatic Attendance System Using Image processing", The International Journal Of Engineering And Science (IJES), vol. 4, issue 11

[4] T Muni Reddy1, V Prasad1, N V Ramanaiah, "Face Recognition Based Attendance Management System By Using Embedded Linux", IJERST, vol. 4, no, 2, may 2014.

[5] Abhishek Jha, "Class Room Attendance System using Facial Recognition System", The InternationalJournal of Mathematics, Science, Technology and Management. (ISSN: 2319-8125) Vol. 2 Issue 3.

[6] Vikram K and S. Padmavathi. "Facial parts detection using Viola Jones algorithm". Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on IEEE, 2017.

[7] Facial recognition system. (2020). Retrieved March 25, 2020, from https://en.wikipedia.org/wiki/Facial_recognition

[8] Arduino Uno. (2020). Retrieved March 25, 2020, from https://en.mwikipedia.org/wiki/Arduino_Uno

[9] MATLAB. (2020). Retrieved April 3, 2020, from https://en.m.wikipedia.org/wiki/MATLAB

11] Rahman, F.H.; Iqbal, A.Y.M.; Newaz, S.S.; Wan, A.T.; Ahsan, M.S. Street parked vehicles based vehicular fog computing: Tcp throughput evaluation and future research direction. In Proceedings of the 2019 21st International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea, 17–20 February 2019;

[12] Wijaya, I.G.P.S.; Husodo, A.Y.; Arimbawa, I.W.A. Real time face recognition using DCT coefficients based face descriptor. In Proceedings of the 2016 International Conference on Informatics and Computing (ICIC), Mataram, Indonesia, 28–29 October 2016;

[13] Ochoa-Dominguez, H.; Rao, K. Discrete Cosine Transform, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2019.

[14] Abdullah, M.A.; Awal, M.A.; Hasan, M.K.; Rahman, M.A.; Alahe, M.A. Optimization of Daily Physical Activity Recognition with Feature Selection. In Proceedings of the 2019 4th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 20–22 December 2019;

[15] Ying, W.; Yongping, Z.; Fang, X.; Jian, X. Analysis Model for Fire Accidents of Electric Bicycles Based on Principal Component Analysis. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 21–24 July 2017;

[16] Li, B. A principal component analysis approach to noise removal for speech denoising. In Proceedings of the 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Hunan, China, 10–11 August 2018;

[17] Hegde, N.; Preetha, S.; Bhagwat, S. Facial Expression Classifier Using Better Technique: FisherFace Algorithm. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018;

[18] Weber, M. The Probabilistic Eigenspace Approach. Available online: www.vision.caltech.edu (accessed on 20 July 2020).

[19] Jin, S.; Lin, Y.; Wang, H. Automatic Modulation Recognition of Digital Signals Based on Fisherface. In Proceedings of the 2017 IEEE International Conference on Software

Quality, Reliability and Security Companion (QRS-C), Prague, Czech Republic, 25–29 July 2017;

[20] Shen, W.; Wang, W. Node Identification in Wireless Network Based on Convolutional Neural Network. In Proceedings of the 2018 14th International Conference on Computational Intelligence and Security (CIS), Hangzhou, China, 16–19 November 2018;

[21] Shalini, K.; Ravikurnar, A.; Vineetha, R.; Aravinda, R.D.; Anand, K.M.; Soman, K. Sentiment Analysis of Indian Languages using Convolutional Neural Networks. In Proceedings of the 2018 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 4–6 January 2018;

[22] Xiao, Y.; Keung, J. Improving Bug Localization with Character-Level Convolutional Neural Network and Recurrent Neural Network. In Proceedings of the 2018 25th Asia-Pacific Software Engineering Conference (APSEC), Nara, Japan, 4–7 December 2018;

[23] Mahajan, N.V.; Deshpande, A.; Satpute, S. Prediction of Fault in Gas Chromatograph using Convolutional Neural Network. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019;

[24] Lee, E.J. Human Motion Recognition Based on Improved 3-Dimensional Convolutional Neural Network. In Proceedings of the 2019 IEEE International Conference on Computation, Communication and Engineering (ICCCE), Fujian, China, 8–10 November 2019;

APPENDIX - I

SOURCE CODE

```
function streamingFaceRecognition(nOfEach,pauseval)
vidObj = webcam; %Default; You can tweak webcam properties
s1 = 0;
s2 = 0;
s3 = 0;
s4 = 0;
s5 = 0;
tcount = 0;
send_mail = 0;
preprocessOpts.matchHistograms = true;
preprocessOpts.adjustHistograms = false;
preprocessOpts.targetForHistogramAndResize = ...
imread('targetFaceHistogram.pgm');
preprocessOpts.targetSize = 100;
mail = 'projectmatlab834@gmail.com';
password = 'Project.1234';
s1_email = 'gontla.dharani@gmail.com';
s2_email = 'saidharani425@gmail.com';
setpref('Internet','SMTP_Server','smtp.gmail.com');
setpref('Internet','E_mail',mail);
setpref('Internet','SMTP_Username',mail);
setpref('Internet','SMTP_Password',password);
props = java.lang.System.getProperties;
props.setProperty('mail.smtp.auth','true');
props.setProperty('mail.smtp.starttls.enable','true');
props.setProperty('mail.smtp.socketFactory.class', 'javax.net.ssl.SSLSocketFactory');
```

```
%props.setProperty('mail.smtp.socketFactory.port','25');

%props.setProperty('mail.smtp.socketFactory.port','465');


%%% DIRECTORY MANAGEMENT:

targetDirectory = fullfile(fileparts(which(mfilename)),'AutoCapturedFaces');

validateCapturedImages = true;

personNumber = 1;

dirExists = exist(targetDirectory,'dir') == 7;

if dirExists

prompt = sprintf('Would you like to:\n\nSTART OVER (Clears Existing Data!!)\nAdd Face(s) to recognition set\nor Use recognition set as is?');

refresh = questdlg(prompt,'Face Recognition Options','START OVER','Add Face(s)','Use as is','START OVER');

refreshOption = find(ismember({'START OVER','Add Face(s)','Use as is'},refresh));

else

mkdir(targetDirectory);

refreshOption = 1;

end

if refreshOption == 1

rmdir(targetDirectory,'s');

mkdir(targetDirectory)

mkdir(fullfile(targetDirectory,filesep,['Person' num2str(1)]))

personNumber = 1;

elseif refreshOption == 2

tmp = dir(targetDirectory);

fcn = @(x)ismember(x,{'.','..'});

tmp = cellfun(fcn,{tmp.name},'UniformOutput',false);

personNumber = nnz(~[tmp{:}])+1;

mkdir(fullfile(targetDirectory,filesep,['Person' num2str(personNumber)]))

elseif refreshOption == 3

% Use as is--no validation of capture!
```

```
validateCapturedImages = false;
elseif isempty(refreshOption)
delete(vidObj)
return
end
%%% FIGURE
fdrFig = figure('windowstyle','normal',...
'name','RECORD FACE UNTIL BEEP; Press <ESCAPE> to Stop',...
'units','normalized',...
'menubar','none',...
'position',[0.2 0.1 0.6 0.7],...
'closerequestfcn',[],...
'currentcharacter','0',...
'keypressfcn',@checkForEscape);
%%% Quality Control Options
%DETECTORS: for upright faces; and for QE, Nose and Mouth
% Note: these seem to be unnecessary, and to cause capture problems.
QC.oneNose = false;
QC.oneMouth = false;
if QC.oneNose
QC.noseDetector = vision.CascadeObjectDetector(...
'ClassificationModel','Nose','MergeThreshold',10);
end
if QC.oneMouth
QC.mouthDetector = vision.CascadeObjectDetector(...
'ClassificationModel','Mouth','MergeThreshold',10);
end
% H,W of bounding box must be at least this size for a proper detection
QC.minBBSize = 30;
% Create face detector
```

```
faceDetector = vision.CascadeObjectDetector('MergeThreshold',10);
% Number of images of each person to capture:
if nargin < 1
nOfEach = 10;
end
%Between captured frames (allow time for movement/change):
if nargin < 2
pauseval = 0.5;
end
% For cropping of captured faces:
bboxPad = 25;
%
captureNumber = 0;
isDone = false;
getAnother = true;
%%% START: Auto-capture/detect/train!!!
RGBFrame = snapshot(vidObj);
frameSize = size(RGBFrame);
imgAx = axes('parent',fdrFig,...
'units','normalized',...
'position',[0.05 0.45 0.9 0.45]);
imgHndl = imshow(RGBFrame);shg;
disp('Esc to quit!')
if ismember(refreshOption,[1,2]) && getAnother && ~isDone
while getAnother && double(get(fdrFig,'currentCharacter')) ~= 27
% If successful, displayFrame will contain the detection box.
% Otherwise not.
[displayFrame, success] = capturePreprocessDetectValidateSave;
if success
captureNumber = captureNumber + 1;
```

```
end
set(imgHndl,'CData',displayFrame);
if captureNumber >= nOfEach
beep;pause(0.25);beep;
queryForNext;
end
end %while getAnother
end
%%% Capture is done. Now for TRAINING:
imgSet = imageSet(targetDirectory,'recursive');
if numel(imgSet) < 2
error('streamingFaceRecognition: You must capture at least two individuals for this to work!');
end
if refreshOption ~= 3
queryForNames;
end
if validateCapturedImages
validateCaptured(imgSet);
end
sceneFeatures = trainStackedFaceDetector(imgSet);
%%% Okay, so now we should have a recognizer in place!!!
figure(fdrFig)
while double(get(fdrFig,'currentCharacter')) ~= 27 && ~isDone
bestGuess = '?';
tcount = tcount+1;
disp(tcount)
if (tcount > 20)
if send_mail == 0
if s1 == 0
sendmail(s1_email,'Mail from Jain University','Hello! This is to inform you that your ward -Person
1 is absent')
```

```
disp('mail sent to Parent-1')

end

if s2 == 0

sendmail(s2_email,'Mail from Jain University','Hello! This is to inform you that your ward -Person
2 is absent')

disp('mail sent to Parent-2')

end

send_mail = 1;

end

end

RGBFrame = snapshot(vidObj);

grayFrame = rgb2gray(RGBFrame);

bboxes = faceDetector.step(grayFrame);

for jj = 1:size(bboxes,1)%#ok

if all(bboxes(jj,3:4) >= QC.minBBSize)

thisFace = imcrop(grayFrame,bboxes(jj,:));

if preprocessOpts.matchHistograms

thisFace = imhistmatch(thisFace,...

preprocessOpts.targetForHistogramAndResize);

end

if preprocessOpts.adjustHistograms

thisFace = histeq(thisFace);

end

thisFace = imresize(thisFace,...

size(preprocessOpts.targetForHistogramAndResize));

%tic;

bestGuess = myPrediction(thisFace,sceneFeatures,numel(imgSet));

tdt = datestr(now,'HH:MM:SS');

ttime = datestr(now,'dd/mm/yyyy');

%disp('ID number of PIC');

id=sprintf('%1.0f',bestGuess);
```

```
%disp(id);

if( id == '1')

if s1 == 0

s1 = 1;

disp('ID 1 Present');

disp(datestr(now));


fid = fopen('Attendance.txt', 'a'); % This portion of code writes the number plate

fprintf(fid,'Person 1 ID:%s - %s - %s\n',id,ttime,tdt);  % to the text file, if executed a notepad file
with the

end

end

if( id == '2')

if s2 == 0

s2 = 1;

disp('ID 2 Present');

disp(datestr(now));

fid = fopen('Attendance.txt', 'a'); % This portion of code writes the number plate

fprintf(fid,'Person 2 ID:%s - %s - %s\n',id,ttime,tdt);  % to the text file, if executed a notepad file
with the

end

end

if bestGuess == 0

bestGuess = '?';

else

bestGuess = imgSet(bestGuess).Description;

end

%tPredict = toc

RGBFrame = insertObjectAnnotation(RGBFrame, 'rectangle', bboxes(jj,:),
bestGuess,'FontSize',48);

end

end
```

```
imshow(RGBFrame,'parent',imgAx);drawnow;
title([bestGuess '?'])
end %while
%%% Clean up:
delete(vidObj)
release(faceDetector)
delete(fdrFig)


%%
% NESTED SUBFUNCTIONS
function [displayFrame, success, imagePath] = ...
capturePreprocessDetectValidateSave(varargin)
% CAPTURE
RGBFrame = snapshot(vidObj);
% Defaults:
displayFrame = RGBFrame;
success = false;
imagePath = [];
grayFrame = rgb2gray(RGBFrame);
% PREPROCESS
if preprocessOpts.matchHistograms
grayFrame = imhistmatch(grayFrame,...
preprocessOpts.targetForHistogramAndResize); %#ok<*UNRCH>
end
if preprocessOpts.adjustHistograms
grayFrame = histeq(grayFrame);
end
preprocessOpts.targetSize = 100;
% DETECT
```

```
bboxes = faceDetector.step(grayFrame);
% VALIDATE
if isempty(bboxes)
return
end
if size(bboxes,1) > 1
disp('Discarding multiple detections!');
return
end
if any(bboxes(3:4) < QC.minBBSize)
disp('Bounding box is too small!');
return
end
% On-the-fly QC!
if QC.oneMouth
mouthBox = QC.mouthDetector.step(grayFrame);
if size(mouthBox,1) ~= 1
%disp('Detected face failed MOUTH QE, and was discarded.')
return
end
end
if QC.oneNose
noseBox = QC.noseDetector.step(grayFrame);
if size(noseBox,1) ~= 1
%disp('Detected face failed NOSE QE, and was discarded.')
return
end
end
% If we made it to here, the capture was successful!
success = true;
```

```
% Update displayFrame

displayFrame = insertShape(RGBFrame, 'Rectangle', bboxes,...

'linewidth',4,'color','cyan');

% SAVE

% Write to personN directory

bboxes = bboxes + [-bboxPad -bboxPad 2*bboxPad 2*bboxPad];

% Make sure crop region is within image

bboxes=[max(bboxes(1),1)max(bboxes(2),1)min(frameSize(2),bboxes(3))
min(frameSize(2),bboxes(4))];

faceImg = imcrop(grayFrame,bboxes);

minImSize = min(size(faceImg));

thumbSize = preprocessOpts.targetSize/minImSize;

faceImg = imresize(faceImg,thumbSize);

% if matchHistograms

% faceImg = imhistmatch(faceImg,targetForHistogramAndResize); %#ok<*UNRCH>

% end

%Defensive programming, since we're using floating arithmetic

%and we need to make sure image sizes match exactly:

sz = size(faceImg);

if min(sz) > preprocessOpts.targetSize

faceImg = faceImg(1:preprocessOpts.targetSize,1:preprocessOpts.targetSize);

elseif min(sz) < preprocessOpts.targetSize

% Not sure if we can end up here, but being safe:

faceImg = imresize(faceImg,[preprocessOpts.targetSize,preprocessOpts.targetSize]);

end

imagePath = fullfile(targetDirectory,...

['Person' num2str(personNumber)],filesep,['faceImg' num2str(captureNumber) '.png']);

imwrite(faceImg,imagePath);

pause(pauseval)

end %captureAndSaveFrame
```

```
function checkForEscape(varargin)
if double(get(gcf,'currentcharacter'))== 27
isDone = true;
end
end %checkForEscape

function queryForNames
prompt = {imgSet.Description};
dlg_title = 'Specify Names';
def = prompt;
renameTo = inputdlg(prompt,dlg_title,1,def);
subfolders = pathsFromImageSet(imgSet);
for ii = 1:numel(renameTo)
subf = subfolders{ii};
fs = strfind(subf,filesep);
subf(fs(end)+1:end) = '';
subf = [subf,renameTo{ii}];%#ok
if ~isequal(subfolders{ii},subf)
movefile(subfolders{ii},subf);
end
end
imgSet = imageSet(targetDirectory,'recursive');
end %queryForNames
function queryForNext
beep
captureAnother = questdlg(['Done capturing images for person ', num2str(personNumber), '. Capture
Another?'],...
'Capture Another?','YES','No','YES');
if strcmp(captureAnother,'YES')
personNumber = personNumber + 1;
captureNumber = 0;
```

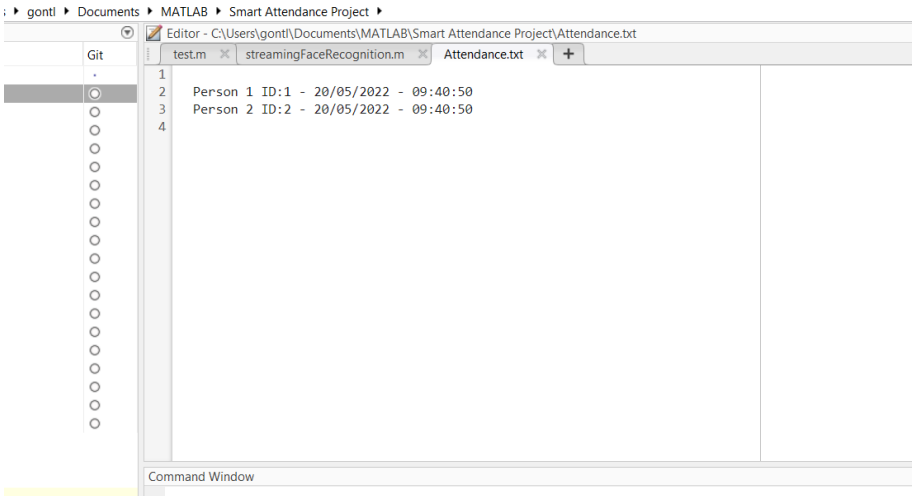
```
mkdir(fullfile(targetDirectory,filesep,['Person' num2str(personNumber)]))
else
getAnother = false;
end
end %queryForNext
function validateCaptured(imgSet)
%assignin('base','imgSet',imgSet)
for ii = 1:numel(imgSet)
nImages = imgSet(ii).Count;
nCols = ceil(sqrt(nImages));
nRows = ceil(sqrt(nImages));
[hobjpos,hobjdim] = distributeObjects(nCols,0.025,0.95,0.025);
[vobjpos,vobjdim] = distributeObjects(nRows,0.9,0.2,0.1);
f = togglefig('Validation',true);
set(f,'windowstyle','normal')
drawnow
btn = gobjects(nImages,1);
ax = btn;
for jj = 1:nImages %#ok
ax(jj) = axes('units','normalized',...
'position',[hobjpos(rem(jj-1,nCols)+1) vobjpos(ceil(jj/nCols)) hobjdim vobjdim]);
imshow(imread(imgSet(ii).ImageLocation{jj}));
if jj == 2
title(imgSet(ii).Description)
end
btn(jj) = uicontrol('style','checkbox',...
'string','Discard',...
'units','normalized',...
'value',0,...
'userdata',jj,...
```

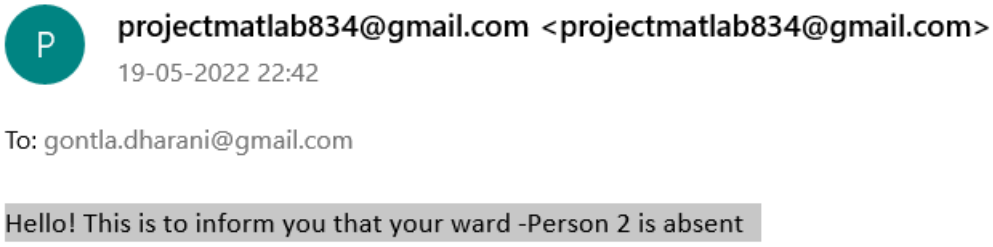
```
'Position',[hobjpos(rem(jj-1,nCols)+1) vobjpos(ceil(jj/nCols))-0.075 hobjdim 0.075]);
end
uicontrol('style','pushbutton',...
'string','Continue',...
'units','normalized',...
'position',[0.025 0.025 0.95 0.1],...
'callback',@registerSelection);
uiwait(f)
end

function registerSelection(varargin)
togglefig('Validation')
btnvals = find(cell2mat(get(btn,'value')));
if ~isempty(btnvals)
confirmDeletion = questdlg(sprintf('Delete the selected %d image(s) from the collection of %s images?', ...
numel(btnvals),imgSet(ii).Description),...
'Confirm Deletion','DELETE','No','DELETE');
if strcmp(confirmDeletion,'DELETE')
for kk = 1:numel(btnvals)
imgSet = removeImageFromImageSet(imgSet,imgSet(ii).ImageLocation{btnvals(kk)});
% Note: deleting images causes problems with the imageset object
% delete(imgSet(ii).ImageLocation{btnvals(kk)});
end
end
end
delete(f);
end %registerSelection (subfunction of validateCaptured)
end %validateCaptured
end
```

APPENDIX-II

DATASHEETS





DETAILS OF PAPER PUBLICATION (ALONG WITH PAPER)

Smart Attendance Using Face Recognition

Bhavana. L Eakila Ajith Bhargav R. Sukumar G Sai Dharani N Lohithas Reddy

Abstract: A face recognition system is one of the biometric information processes, its applicability is easier and working range is larger than others, i.e.; fingerprint, iris scanning, signature, etc. A face recognition system is designed, implemented and tested. The system uses a combination of techniques in two topics; face detection and recognition. The face detection is performed on live acquired images without any application field in mind. Processes utilized in the system are white balance correction, skin like region segmentation, facial feature extraction and face image extraction on a face candidate. Then a face classification method that uses Feedforward Neural Network is integrated in the system. The system is tested with a database generated in the laboratory with 26 people. The tested system has acceptable performance to recognize faces within intended limits. System is also capable of detecting and recognizing multiple faces in live acquired images.

1.Introduction: During this era of technology and automation we are still using the same old ways of attendance management. The most important thing in the classroom/office is attendance which is directly linked to the academic performance of the students/employees. Recently, some students are husky with better skills during Lectures only when there is massive classroom control (Research Gale, 2018)[1]. The more efficient the attendance system the mare is participation and learning. In the past we were using techniques like roll numbering calling and signing against a particular roll number. These methods carry a high chance of proxy and are time consuming. We came across the idea of automating this process through modern day technologies to get a well maintained and disciplined attendance. Facial recognition systems along with suitable hardware and software will help meet the goals of this project[2]. Facial recognition system is derived incovalion of image processing Image processing deals with the extraction of needy data that can be related to digital Image and in technology advancement it plays a unique role. Our core focus will be on receiving digital Images and then making use of programs and algorithms to get useful Information out of it. As

the pictorial information is for the image processing work on it and make it esetl human interpretation[3]. That information from image processing will play a great role and help in various walks of life where it could be implemented. The applications of image processing are vast and can be applied in most scenarios where imaging data could be related to pre-determined algorithms. It was an advanced application of image processing and also is the core basis for our project.

Our facial structure was a typical example of a multidimensional structure and needed some recognition from advanced computational analysis[4]. The project aims at designing an intelligent security system for recording the attendance in colleges, firms etc..,. The proposed system makes use of Face Recognition technology for recording the attendance. The system records the attendance along with enter and exit date and time. The Intelligent system proposed makes use of MATLAB software to achieve the task[5]. We are living in a world where everything is automated and linked online. The internet of things, image processing and machine learning are evolving day by day. Many systems have been completely changed due to this evolve to achieve more accurate results. The attendance system is a typical example of this transition, starting from the traditional signature on a project sheet to face recognition. In this project we are proposing an automatic attendance system which can be used in every organization to mark the attendance of employees. The main application of Automatic attendance system is seen in teaching institutions, where the attendance of students has to be regularly monitored on daily basis. The method developed provides a secure and effective attendance record. Automatic attendance system uses MATLAB[6]

2.Methadology:

A. Enrollment Phase

1)ImageAcquisition: For Image acquisition, camera is used to capture an image of the employees working in an organization.

2) Face Detection and Face cropping: For face detection cascading algorithm is used. Further face cropping is done using imcrop (image) command that separates the facial area from the rest of the background image. This faces is stored in the database.

3) Feature Extraction: Feature extraction is done by using linear binary pattern algorithm. Feature extraction is helpful for face detection and recognition.

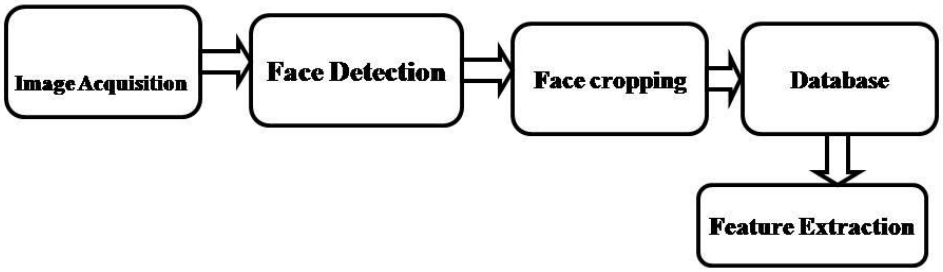
4) Database: Extracted features of employees or persons will be stored in Database.

B. Routine Attendance after Enrollment Phase

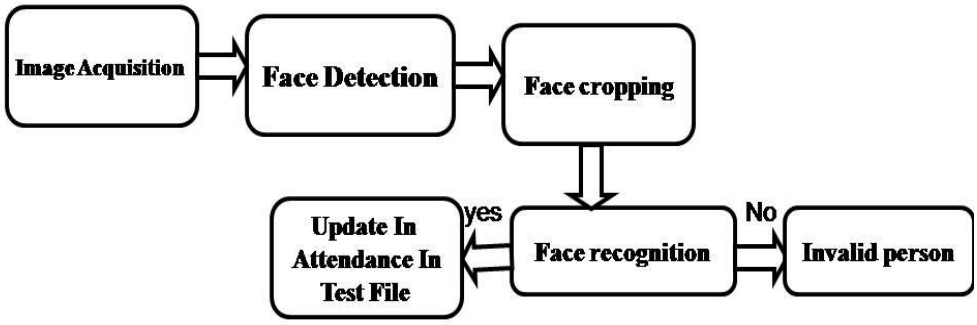
[3] Routine attendance will start after extracting the features of all the images stored in the database. We have followed the same steps as given in enrollment phase i.e. image acquisition, Face detection, and face cropping.

Face recognition is done by using the Local binary pattern by taking 3x3 block a time and comparing central block with surrounding blocks and compare with features stored in the database. If after face recognition face of the person is matched in the database then update the attendance otherwise show the dialog box with invalid person.

ENROLLMENT PHASE



TRAINING STARTS AFTER FEATURE EXTRACTION



Conclusion: The project "SMART ATTENDANCE SYSTEM USING FACE RECOGNITION" was designed an intelligent security system for recording the attendance in colleges, firms etc..,. The proposed system makes use of Face Recognition technology for recording the attendanceusing MATLAB software. The topic chosen bythe groupwas a digitalsystemon facialdetection and facial recognition. At the start of the project, the group'sobjectiveswere torepresent adigitalsystemina simple yet effective way, to successfully simulate a facial detection and recognition program, to learn more about the functions that MATLAB has to offer, and to increase their knowledge on the various applicationsthat digital communication has. The eyes ofthe group were opened to the various possible topics that they could tackle and the effects that each topic hasin the world that we live in today. There search of the group led to the realization about how the modernization of technology brings even more possibilities and ideas for the future of digital communications.

References:

Research papers:

[1] Muhammad Fuzail, Hafiz Muhammad Fahad Nouman, Muhammad Omer Mushtaq, Binish Raza, Awais Tayyab, and Muhammad Waqas Talib, "Face Detection System for Attendance of Class' Students", International Journal Of Multidisciplinary Sciences And Engineering, vol. 5, no. 4, April 2014

[2]Adrian Rhesa Septian Siswanto, Anto Satriyo Nugroho, MaulahikmahGalinium,"Implementation of Face Recognition Algorithm for Biometrics Based Time Attendance System", IEEE Conference.

[3] Aziza Ahmedi , 2Dr Suvarna Nandyal, "An Automatic Attendance System Using Image processing", The International Journal Of Engineering And Science (IJES), vol. 4, issue 11

[4] T Muni Reddy1, V Prasad1, N V Ramanaiah, "Face Recognition Based Attendance Management System By Using Embedded Linux", IJERST, vol. 4, no, 2, may 2014.

[5] Abhishek Jha, "Class Room Attendance System using Facial Recognition System", The InternationalJournal of Mathematics, Science, Technology and Management. (ISSN: 2319- 8125) Vol. 2 Issue 3.

[6] Vikram K and S. Padmavathi. "Facial parts detection using Viola Jones algorithm". Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on IEEE, 2017.

[7] Facial recognition system. (2020). Retrieved March 25, 2020, from https://en.wikipedia.org/wiki/Facial_recognition

[8] Arduino Uno. (2020). Retrieved March 25, 2020, from https://en.mwikipedia.org/wiki/Arduino_Uno

[9] MATLAB. (2020). Retrieved April 3, 2020, from https://en.m.wikipedia.org/wiki/MATLAB [10] Rahman, F.H.; Newaz, S.S.; Au, T.W.; Suhaili, W.S.; Mahmud, M.P.; Lee, G.M. EnTruVe: ENergy and TRUst-aware Virtual

Machine allocation in VEhicle fog computing for catering applications in 5G. Future Gener. Comput. Syst. 2021, 126, 196–210.

[11] Rahman, F.H.; Iqbal, A.Y.M.; Newaz, S.S.; Wan, A.T.; Ahsan, M.S. Street parked vehicles based vehicular fog computing: Tcp throughput evaluation and future research direction. In Proceedings of the 2019 21st International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea, 17–20 February 2019;

[12] Wijaya, I.G.P.S.; Husodo, A.Y.; Arimbawa, I.W.A. Real time face recognition using DCT coefficients based face descriptor. In Proceedings of the 2016 International Conference on Informatics and Computing (ICIC), Mataram, Indonesia, 28–29 October 2016;

[13] Ochoa-Dominguez, H.; Rao, K. Discrete Cosine Transform, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2019.

[14] Abdullah, M.A.; Awal, M.A.; Hasan, M.K.; Rahman, M.A.; Alahe, M.A. Optimization of Daily Physical Activity Recognition with Feature Selection. In Proceedings of the 2019 4th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 20–22 December 2019;

[15] Ying, W.; Yongping, Z.; Fang, X.; Jian, X. Analysis Model for Fire Accidents of Electric Bicycles Based on Principal Component Analysis. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 21–24 July 2017;

[16] Li, B. A principal component analysis approach to noise removal for speech denoising. In Proceedings of the 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Hunan, China, 10–11 August 2018;

[17] Hegde, N.; Preetha, S.; Bhagwat, S. Facial Expression Classifier Using Better Technique: FisherFace Algorithm. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018;

[18] Weber, M. The Probabilistic Eigenspace Approach. Available online: www.vision.caltech.edu (accessed on 20 July 2020).

[19] Jin, S.; Lin, Y.; Wang, H. Automatic Modulation Recognition of Digital Signals Based on Fisherface. In Proceedings of the 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Prague, Czech Republic, 25–29 July 2017;

[20] Shen, W.; Wang, W. Node Identification in Wireless Network Based on Convolutional Neural Network. In Proceedings of the 2018 14th International Conference on Computational Intelligence and Security (CIS), Hangzhou, China, 16–19 November 2018;

[21] Shalini, K.; Ravikurnar, A.; Vineeth, R.; Aravinda, R.D.; Anand, K.M.; Soman, K. Sentiment Analysis of Indian Languages using Convolutional Neural Networks. In Proceedings of the 2018 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 4–6 January 2018;

[22] Xiao, Y.; Keung, J. Improving Bug Localization with Character-Level Convolutional Neural Network and Recurrent Neural Network. In Proceedings of the 2018 25th Asia Pacific Software Engineering Conference (APSEC), Nara, Japan, 4–7 December 2018;

[23] Mahajan, N.V.; Deshpande, A.; Satpute, S. Prediction of Fault in Gas Chromatograph using Convolutional Neural Network. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019;

Faculty reference:

[24] S Amulya,: Pilli Siva Gowtham Reddy,: SA Hariprasad,; MP Sunil,. FACE DETECTION AND RECOGNITION FOR AUTOMATIC ATTENDANCE SYSTEM. International Education and Research Journal, 2018;

INFORMATION REGARDING STUDENTS

STUDENT NAME	EMAIL ID	PERMANENT ADDRESS	PHONE NUMBER	PLACEMENT DETAILS	PHOTOGRAPHY
Bhavana. L	bhavanalshetty@gmail.com	Opposite keb quatrus Main road, tk halli Mandya, Karnataka.	8861582868	Placed	
Eakila Ajith Bhargav	ajith7.ab@gmail.com	21/223, brahman street, oldtown, Anantapur, Andhra Pradesh (515001)	9515231579	Placed	
Gontla Sai Dharani	gontla.dharani@gmail.com	Sri vasavi enclave, Sai Nagar, Giddalur Andhra Pradesh, pincode:523357	7396154546	placed	
Nagarimadugu Lohithas Reddy	nlohithas9@gmail.com	9-87/2, mns street, oldpet, palamaner, chittoor dist, andhra pradesh,517408	7207872215	Placed	

BATCH PHOTOGRAPH ALONG WITH GUIDE

Dr. R. Sukumar

HOD

Bhavana. L

18BTREC010

Eakila Ajith Bhargav

18BTREC018

Gontla Sai Dharani

18BTREC019

Nagarimadugu Lohithas Reddy

18BTREC046