VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY ENGINEERING

MTECH [INTEGRATED] SOFTWARE AND ENGINEERING

BIOMETRIC SYSTEMS (SWE1015)

GROUP – 8

FINAL REVIEW (FALL 2019-20)

FACE RECOGNITION SYSTEM FOR EMPLOYEE ATTENDANCE

BIOMETRIC SYSTEMS

GROUP MEMBERS:-

SNO	REG NO	NAME
1	16MIS0010	R MOHAN RAJ
2	16MIS0121	T PREETHI
3	16MIS0185	G MEGHANA
4	16MIS0433	R MANO BALA
5	16MIS0224	T HARI KRISHNAN

SUBMITTED

TO

PROF:- RAMYA. G

SLOT: - C1+TC

ABSTRACT:

This project implements a face recognition system for employee which first detects the face and then identifies the particular person by comparing the detected face with image database. It will help to maintain a strong security in the local database. Firstly the data sets are created and trained them with the help of algorithms which undergoes feature extraction, normalization and then preprocessed. The trained data sets are compared with the input facial image, it first detects and recognizes the person and displays the name of the personnel and tells weather the person is valid or not.

Keywords:

Face recognition, datasets, database, trained database, test database.

INTRODUCTION:

Now a day's most of the corporate companies and educational institutions are concerned about the regularity of the employees and this is mainly to know the timings of the employees and for surveillance. There are various ways by which one has its unique identity such has finger print, iris, retina, face scanning etc. These biological features are different for different people and hence can be used as their identification for precisely. One of the system that can be used for the identification of a person uniquely is face recognition system. It captures the image of the face and try to recognize the different features available on the face such has forehead lining or shape of the eyes, lips and nose. These features are then stored as the identification of the person and it is stored in the database and next time when the same persons face is captured the person can be searched in the database using the various algorithm. In our project face recognition system a person can be identified.

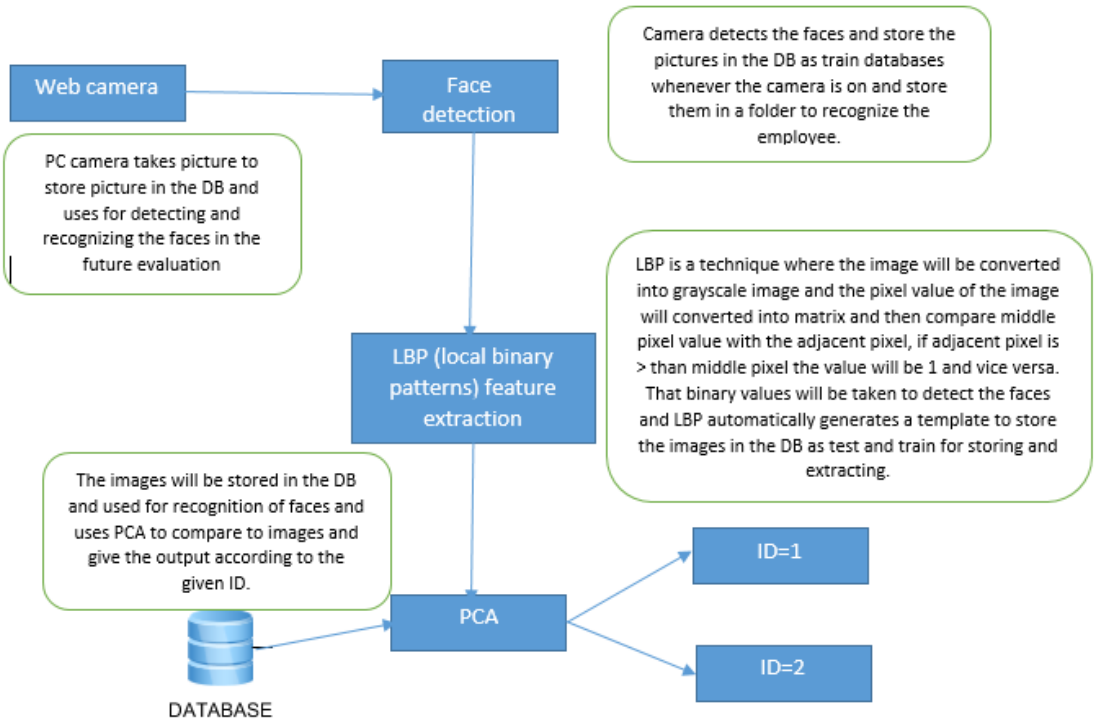
OBJECTIVES:

Our primary goal is to help the lecturers, improve and organize the process of track and manage student attendance and absenteeism. Additionally, we seek to:

- Provides a valuable attendance service for both teachers and students.
- Reduce manual process errors by provide automated and a reliable attendance system uses face recognition technology.
- Increase privacy and security which student cannot presenting himself or his friend while they are not.
- Produce monthly reports for lecturers.
- Flexibility, Lectures capability of editing attendance records.
- Calculate absenteeism percentage and send reminder messages to students.

In this project we aim to build an Attendance marking system with the help of facial recognition owing to the difficulty in the manual as well as other traditional means of attendance system.

ARCHITECTURE:



LITERATURE SURVEY:

1. Chen, L. F., Liao, H. Y. M., Ko, M. T., Lin, J. C.

In this paper they used new LDA based Face Recognition System. Linear discriminant analysis (LDA) is one of the most popular linear projection techniques for feature extraction. The major drawback of applying LDA is that it may encounter the small sample size prob- lem. In this paper, they propose a new LDA-based technique which can solve the small sample size problem. They also prove that the most expressive vectors derived in the null space of the within-class scatter matrix using principal component analysis (PCA) are equal to the optimal discriminant vectors derived in the original space using LDA. The experimental results show that the new LDA process improves the performance of a face recognition system significantly. Many classic and contemporary face recognition algorithms work well on public data sets, but degrade sharply when they are used in a real recognition system. This is mostly due to the difficulty of simultaneously handling variations in illumination, image misalign- ment, and occlusion in the test image. In this paper they consider a scenario where the training images are well controlled and test im- ages are only loosely controlled. They propose a conceptually simple face recognition system that achieves a high degree of robustness and stability to illumination variation, image misalignment, and partial occlusion. The system uses tools from sparse representation to align a test face image to a set of frontal training images. The region of attraction of our alignment algorithm is computed empirically for public face data sets such as Multi-PIE.

2. Wagner, A., Wright, J., Ganesh, A., Zhou, Z., Mobahi, H.

In this paper they used eigenfaces for the Face Recognition. An approach to the detection and identification of human faces is pre- sented, and a working, near-real-time face recognition system which tracks a subject's head and then recognizes the person by comparing characteristics of the face to those of known individuals is described. This approach treats face recognition as a two-dimensional recogni- tion problem, taking advantage of the fact that faces are normally upright and thus may be described by a small set of 2-D character- istic views. Face images are projected onto a feature space ('face space') that best encodes the variation among known face images. The face space is defined by the 'eigenfaces', which are the eigenvec- tors of the set of faces; they do not necessarily correspond to isolated features such as eyes, ears, and noses. The framework provides the ability to learn to recognize new faces in an unsupervised manner.

3. Turk, M. A., Pent-land, A. P. (1991, June). Face recognition using eigenfaces

In this paper they present a system for recognizing human faces from single images out of a large database with one image per per-

son. The task is difficult because of image variation in terms of position, size, expression, and pose. The system collapses most of this variance by extracting concise face descriptions in the form of image graphs. In these, fiducial points on the face (eyes, mouth etc.) are described by sets of wavelet components (jets). Image graph ex- traction is based on a novel approach, the bunch graph, which is constructed from a small set of sample image graphs. Recognition is based on a straight-forward comparison of image graphs.They report recognition experiments on the FERET database and the Bochum database, including recognition across pose.

4. Wiskott, L., Fellous, J. M., Krger, N., Von Der Mals- burg,

In this paper a new method to locate human faces in a com- plex background is proposed. This system utilizes a hierarchical knowledge-based method and consists of three levels. The higher two levels are based on mosaic images at different resolutions. In the lotheyr level, an improved edge detection method is proposed. In this research the problem of scale is dealt with, so that the system can locate unknown human faces spanning a wide range of sizes in a complex black-and-white picture. Some experimental results are given.

A recognition system for identifying members of an audience, the system including an imaging system which generates an image of the audience; a selector module for selecting a portion of the generated image; a detection means which analyzes the selected image por- tion to determine whether an image of a person is present; and a recognition module responsive to the detection means for determin- ing whether a detected image of a person identified by the detection means resembles one of a reference set of images of individuals.

5. Yang, G., Huang, T. S. (1994). Human face detection

In this paper they investigate the performance of a technique for face recognition based on the computation of 25 local autocor- relation coefficients. They use a large database of 11,600 frontal facial images of 116 persons, organized in training and test sets, for evaluation. Autocorrelation coefficients are computationally inex-

pensive, inherently shift-invariant and quite robust against changes in facial expression. They focus on the difficult problem of recog- nizing a large number of known human faces while rejecting other, unknown faces which lie quite close in pattern space. A multiresolu- tion system achieves a recognition rate of 95, while falsely accepting only 1.5 of unknown faces. In this work they describe experiments with eigenfaces for recognition and interactive search in a large-scale face database. Accurate visual recognition is demonstrated using a database of O (103) faces. The problem of recognition under gen- eral viewing orientation is also examined. A view-based multiple- observer eigenspace technique is proposed for use in face recognition under variable pose. In addition, a modular eigenspace description technique is

used which incorporates salient features such as the eyes, nose and mouth, in an eigenfeature layer. This modular rep- resentation yields higher recognition rates as well as a more robust framework for face recognition. An automatic feature extraction technique using feature eigen templates is also demonstrated.

PAPER 6:

TITLE OF THE PAPER AND YEAR:

Face Recognition based Automated Attendance Management System using Hybrid Classifier(2017)

ALGORITHAM /TECHNIQUE USED:

AAMS, Viola-Jones algorithm, PCA and LBP classifier.

PROPOSED WORK:

In the proposed work, a high definition camera is placed on the top center of the classroom as shown in Fig- 1. We capture the student data and the different orientation of student's face that will be stored in the server system.

FUTURE DEVELOPMENT:

The proposed system improves the performance of existing attendance management systems in the following ways:

i) Automatic tracking of the records of the students

ii) Minimizing the manual labor and pressure on the lecturers for accurate marking of the attendance

iii) Minimizing the time required for marking attendance and maximizing the time required for actual teaching process

iv) Increase the efficiency of the overall system Improving the security

CONCLUSION:

To eliminate the manual labor involved in recording attendance, an automated Attendance Management System (AAMS) based on hybrid classifier face detection and face recognition techniques is proposed. The modified Viola-Jones algorithm and PCA and LDB classifier technique together are used for face detection and recognition. We Experiments carried out set of experiments in the classroom by varying number of students.

PAPER 7:-

TITLE OF THE PAPER AND YEAR:

FACE RECOGNITION BASED ATTENDANCE MARKING SYSTEM(2014)

ALGORITHAM /TECHNIQUE USED:

SURF AND FAST FACE DETECTION ALGORITHM

Statistical tools such as Linear Discriminant Analysis (LDA), Principal

Component Analysis (PCA), Kernel Methods, and Neural Networks, Eigen-faces have been

used for construction of face templates.

PROPOSED WORK:

The system consists of a camera that captures the images of the employee and sends it to the

image enhancement module. After enhancement the image comes in the Face Detection and

Recognition modules and then the attendance is marked on the database server. At the time of enrolment, templates of face images of individual employees are stored in the Face database. Here all the faces are detected from the input image and the algorithm compares them one by one with the face database.

CONCLUSION:

The efficient and accurate method of attendance in the office environment that can replace the old manual methods. This method is secure enough, reliable and available for use. No need for specialized hardware for installing the system in the office. It can be constructed using a camera and computer.

PAPER-8:

TITLE OF THE PAPER AND YEAR:

Face Recognition based Attendance Management System using Machine Learning (2018)

ALGORITHAM /TECHNIQUE USED

Face Recognition, Deep Learning, Python, Portable Document Format.

PROPOSED WORK

The project aims to design and implement a system which is less sensitive to Illumination, is rotation invariant, scale invariant and robust enough to be implemented in practical applications.

The project has two main parts:

[A] Development of Face Recognition System.

[B] Development of Attendance System.

Face recognition is achieved using machine learning and the basic pipeline used for it is as follows:

1. Finds face in an image.
2. Analyses facial features.
3. Compares against known faces and makes a prediction.

CONCLUSION:

The purpose of reducing the errors that occur in the traditional attendance taking system has been achieved by implementing this automated attendance system. In this paper, face recognition system have been presented using deep learning which exhibits robustness towards recognition of the users with accuracy of 98.3%.

PAPER 9:-

TITLE OF THE PAPER AND YEAR:

Attendance System based on Face Recognition (2018)

ALGORITHAM /TECHNIQUE USED:

Query Image, Bit- Byte Conversion Methods. Machine learning techniques

CONCLUSION:

It saves time and effort, especially if it is a lecture with huge number of students. The complete system is implemented in MATLAB. This attendance system shows the use of facial recognition techniques for the purpose of student attendance and for the further process this record of student can be used in exam related issues.

PAPER 10:

TITLE OF THE PAPER AND YEAR:-

Face Recognition for E-Attendance for Student and Staff (2017)

ALGORITHAM /TECHNIQUE USED:-

PCA (principle component Analysis)

PROPOSED WORK:

The database contains the record of all of the students in each class. If image matched with database image then student with that face is marked as present. The whole student attendance record will be maintained by Mail Server.

CONCLUSION:

The system automatically updates the attendance of the individuals and mark present/absent for them in an excel sheet. The excel sheet used to update attendance is overwritten every time the program is executed. Hence the user has to save the current attendance in the memory in order to use it for future reference.

11**. Nefian, A. V., Hayes, M. H. (1998, May).**

In this paper, a method for student attendance system in class- room using face recognition technique by using discrete wavelet transforms and discrete cosine transform to extract the features of students face which is followed by use of Radial Basis Function for classifying the facial components.

12. Face Recognition: A Literature Survey W. ZHAO Sarnoff Corporation R. CHELLAPPA University of Maryland P. J. PHILLIPS National Institute of Standards and Technology AND A. ROSENFELD University of Maryland.

This paper provides an up-to-date critical survey of still- and video-based face recognition research. There are two underlying motivations for us to write this survey paper: the first is to provide an up-to-date review of the existing literature, and the second is to offer some insights into the studies of machine recognition of faces. To provide a comprehensive survey, we not only categorize existing recognition techniques but also present detailed descriptions of representative methods within each category. In addition, relevant topics such as psychophysical studies, system evaluation, and issues of illumination and pose variation are covered.

13. A Literature Review on Face Recognition System Avleen Bansal Department of Computer Science Gateway Institute of Engineering and Technology (GIET), Deenbandhu Chhotu Ram University of Science and Technology (DCRUST), Sonepat.

Face recognition is a biometric system used to identify or verify a person from a digital image. Face recognition is undoubtedly an interesting research area, growing in importance in recent

years, due to its applicability as a biometric system in commercial and security applications. These systems could be used to prevent unauthorized access or fraudulent use of ATMs, cellular phones, smart cards, desktop PCs, workstations, and computer networks. The appealing characteristic of a face recognition system is that, differently from fingerprint or iris biometric systems, it represents a not invasive control tool. In this paper we provide literature review of various face recognition system.

14. Brunelli, R., Poggio, T. (1993). Face recognition: Features versus templates

In this paper, the authors proposed the concept of implementa- tion of class-room attendance system based on face recognition in 2014. In this approach, they used face detection and face recognition system. The face detection differentiates face parts from non-face parts and is therefore essential for accurate attendance. The face recognition for marking the student's attendance uses supervised method LDA. The Raspberry pi module is used for face detection recognition.

15. Brunelli, R., Poggio, T. (1993). Face recognition: Features versus templates

In this paper, the author proposed that different types of face detection for detecting faces in different pose .Detecting face in dif- ferent pattern based on techniques. Basic pattern for detecting face is nose, eyes, hair, ears and some time it based on tone of skin. Face

detection is detecting face based on location of face and presences of face in images .Different types of detecting the face techniques they are Ada-Boost Algorithm for Face Detection, Viola Jones Face De- tection Algorithm, SMQT Features and SNOW Classifier Method, Local Binary Pattern (LBP). Each have advantages and disadvan- tages discussed in that paper.

In this paper, the authors proposed that recognition face using hog features and pca algorithms. By applying 0recognition algo- rithm to cropped faces images from that we get similarity b/w taken image and database image. In this paper PAC algorithm used for face detection and recognition. In this paper, the authors shows that face recognition of facial of different person or student .from recognition attendances is upload to database using face detection and recognition of student or workers. From this manual work is decrease by human and automatically attendance system based on faces process done.

16**. Samuel Lukas , Aditya Rama Mitra**

This paper presents a novel and efficient facial image represen- tation based on local binary pattern (LBP) texture features. The face image is divided into several regions from which the LBP fea- ture distributions are extracted and concatenated into an enhanced feature vector to be used as a face descriptor. The performance of the proposed method is assessed in the face recognition problem un- der different challenges. Other applications and several extensions are also discussed The work presented in this paper focuses on the use of hidden Markov models for face recognition. A new method based on the extraction of 2D-DCT feature vectors is described, and the recognition results are compared with other face recogni- tion approaches. The method introduced reduces significantly the computational complexity of previous HMM-based face recognition system, while preserving the same recognition rate.

17. Face Recognition Techniques - An evaluation Study (Department of Management Information System, Applied Science University,)

Face Recognition Based on Principal Component Analysis. Principal Component Analysis (PCA) is known as algorithM That used in face recognition. The basic idea in PCA is to determine a vector of much lower dimension that best approximates in some sense a given data vector, thus, in face recognition it takes an s-dimensional vector representation of each face in a training set of images as input, and determines a t- dimensional subspace whose basis vector is maximum corresponding to the original image, The dimension of this new subspace is lower than the original one (t <<s). if the original image elements are considered as random variables, then the principal corresponding to the large Eigen values of the correlation matrix and error minimization is done in a least-squares sense(Qing chen, Xiaoli Yang, jiying Zhao) .

18. REAL TIME FACE RECOGNITION USING ADBOOST IMPROVED PCA ALGORITHM

AUTHORS: K.Susheel Kumar, Shitala Prasad, Vijay Bhaskar Semwal, R C Tripathi

Represent the faces in the database in terms of the vector X.Compute the average face Avg Face and subtract the Avg Face from the vector X. Classify the images based on the number of unique subjects involved. So the number of classes, C, will be the number of subjects who have been imaged.

19. SPARSE REPRESENTATION THEORY AND ITS APPLICATIONS FOR FACE RECOGNITION

AUTHORS:-Yongjiao Wang , Chuan Wang , and Lei Liang

we use some face databases to verify the performance of different face recognition methods We compare face recognition based sparse representation (SR) with the common methods such as nearest neighbor (NN), linear support vector machine (SVM), nearest subspace (NS). In our experiments, PCA is used to reduce the dimensionality of original image vector, and then these low dimension features are as facial feature. We randomly separate each database into two halves. One half was used as the dictionary, and the other half as testing samples After conversion problem, the optimal solution can be solved by a standard linear programming method to obtain. Obviously, if we directly use the original high-dimensional image to construct the training dictionary, with the corresponding equations must be over- determined, but there is the corresponding high computational complexity. To reduce the computational complexity and maintain the sparsity of the solution vector, the original facial image vector is projected by PCA to obtain the low dimension face vector.

20) A Summary of literature review : Face Recognition Kittikhun Meethongjan and Dzulkifli Mohamad Faculty of Computer Science and Information System, University Technology of Malaysia, 81310 Skudai, Johor, Malaysia.

Approach of face recognition aims to detect faces in still image and sequence image from video have many method such as local, global, and hybrid approach. The main problem of face recognition are intensity, illumination, pose, difficult to controlling and large occlusion. In 3D capture creates larger data files per subject which applies significant storage requirements, slow processing, most new devices can be capture in 3D. This is the problem for our future work that want to solving and create accuracy gain for widely accept in 3D face recognition system.

21) Student attendance system in classroom using face recognition technique

Authors: samuel.lukas, aditya.mitra, ririn.desanti, dion.krisnadi

Authentication is one of the significant issues in the era of information system. Among other things, human face recognition (HFR) is one of known techniques which can be used for user authentication. As an important branch of biometric verification, HFR has been widely used in many applications, such as video monitoring/surveillance system, human-computer interaction, door access control system and network security. This paper proposes a method for student attendance system in classroom using face recognition technique by combining Discrete Wavelet Transforms (DWT) and Discrete Cosine Transform (DCT) to extract the features of student's face which is followed by applying Radial Basis Function (RBF) for classifying the facial objects. From the experiments which is conducted by involving 16 students situated in classroom setting, it results in 121 out of 148 successful faces recognition.

22) A Survey on Face Recognition based Students Attendance System

Authors: Binyam Tesfahun Liyew1 , Prasun Hazari2

Face recognition is the detection and identification of humans by the unique characteristics of their Faces. Face recognition technology is the least intrusive and fastest bio-metric technology. It works with the most obvious individual identifier the human face. This research aims at providing a system to automatically record the students' attendance during lecture hours or exam in a hall or room using facial recognition technology instead of the traditional manual methods. The objective of this research is to thoroughly study the field of pattern recognition (facial recognition), which is very important and is used in various applications like identification and detection. And finally, apply this technology to support the student's attendance system. These will help the attendance system to record more efficiently. The proposed system will update the attendance once the students face is match with the template database

23) Real Time Face Recognition System for Time and Attendance Application

Authors: R.Jagadish1 R.Divya2 C.Rengalakshmi3 K.Vidhysree4 T.Ponmeena5

This paper presents an automated system for human face recognition in a real time background for aorganisation to mark the attendance of their employee. So Smart Attendance using Real Time Face Recognition is a real world solution which comes with day to day activities of handling employee. The task is very difficult as the real time background subtraction in an image is still a challenge. To detect real time human face, Principal Component Analysis (PCA) is used to recognize the faces detected with a high accuracy rate. The matched face is then used to mark attendance of the employee once recognition is done, automatically attendance will be updated in an Excel Sheet along with his name, date and time. This project gives much more solutions with accurate results in user interactive manner rather than existing attendance and leave management systems. In addition to that we provide ideal solution to the problem of power wastage is appliance available in the cabin is controlled through face recognition. The ease of deployment is due to wireless mode of communication. A prototype of the controller is implemented, and the experiment results show that the controller can easily and flexibly control the appliance.

24) Implementation of Automated Attendance System using Face Recognition.

Authentication is an issue in computer based communication.Face recognition is widely used in many applications such as system security and door control system.The paper describes how to take student's attendance using face recognition.The face recognition is implemented with the help of Principal Component Analysis (PCA) algorithm.The system will recognize the face of the student and saves the resp ponse in database automatically.The system also includes the feature of retrieving the list of students who are absent in a particular day.

Author: Mathana Gopala Krishnan, Balaji, Shyam Babu

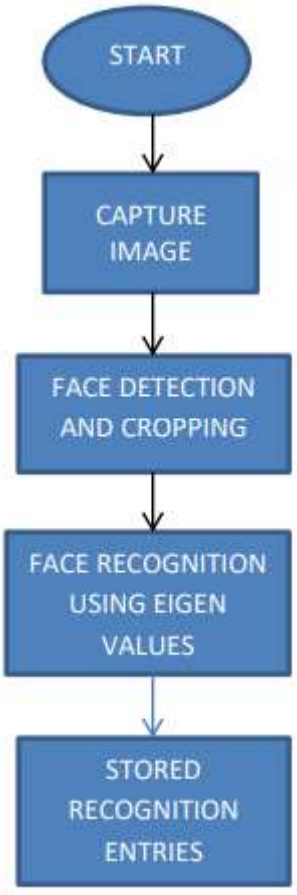
25) Face Recognition: A Literature Survey W. ZHAO Sarnoff Corporation R. CHELLAPPA University of Maryland P. J. PHILLIPS National Institute of Standards and Technology AND A. ROSENFELD University of Maryland.

This paper provides an up-to-date critical survey of still- and video-based face recognition research. There are two underlying motivations for us to write this survey paper: the first is to provide an up-to-date review of the existing literature, and the second is to offer some insights into the studies of machine recognition of faces. To provide a comprehensive survey, we not only categorize existing recognition techniques but also present detailed descriptions of representative methods within each category. In addition, relevant topics such as psychophysical studies, system evaluation, and issues of illumination and pose variation are covered.

MODULES:

1. IMAGE ACQUISITION [TRAINING]
2. PRE PROCESSING
3. EXTRACTION
4. ENROLLMENT
5. DATABASE
6. DETECTION
7. RECOGNITION
8. DISPLAY NAME/PRESENT/ABSENT

IMAGE ACQUISITION:



The first stage of any vision that is the colored image is the image acquisition stage. Captures the images for the datasets and stores it in the database for future inference. The stored datasets are well-trained.

PREPROCESSING:
Clean up the raw data so that it is in the best possible state to make recognition. Data sets acquired are processed in order to normalize in similar ranges. Specify the fixed size for all the images so that it is easy for acquiring accurate features for recognition.

EXTRACTION:
The most important features for recognition from the pre-processed biometric data is extracted. That is in this module we select some nodal points such as : length between eyes, face color, nose length, chin length, distance between chin and neck etc.,.

ENROLLMENT:
The extracted features and information of the faces are enrolled to the system so that it is used when detecting and recognising the face when it appears. Image archive is stored that is the sample faces are stored in the database as a TEMPLATE.

DATABASE In biometric, the database is housed for both enrollment and verification templates of the end users. Used in client-server topology.

DETECTION:
The objective of finding the faces(location anf size) in an image and probably extract them to be used by the face recognition algorithm.

RECOGNITION:
With the facial images already extracted, pre-processed i.e cropped, resized and usually converted to greyscale, the face recognition algorithm is responsible for finding characteristics which best describe the image. Can basically operate on two modes: Verification or authentication of a facial image: compares template with the input facial image of the user which is requiring authentication.(1*1) Identification or facial recognition:compares i/p facial image with dataset to find the user that matches the face.(1*N)

DISPLAY NAME/ABSENT/PRESENT:
Finally after performing all the above processes, it gives the output i.e, it displays the name of the person when placed under the camera.

CODE IMPLEMENTATION AND SCREENSHOTS:

```
function varargout = main(varargin)
% MAIN MATLAB code for main.fig
%      MAIN, by itself, creates a new MAIN or raises the existing
%      singleton*.
%
%      H = MAIN returns the handle to a new MAIN or the handle to
%      the existing singleton*.
%
%      MAIN('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in MAIN.M with the given input arguments.
%
%      MAIN('Property','Value',...) creates a new MAIN or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before main_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to main_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help main

% Last Modified by GUIDE v2.5 03-Nov-2019 23:49:52

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @main_OpeningFcn, ...
                   'gui_OutputFcn',  @main_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
```

```
end
% End initialization code - DO NOT EDIT


% --- Executes just before main is made visible.
function main_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to main (see VARARGIN)

% Choose default command line output for main
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes main wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = main_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% standard size of image is 300 *300
global co
clc
warning off
st = version;
if str2double(st(1)) < 8
   beep
   hx  = msgbox('PLEASE RUN IT ON MATLAB 2013 or Higher','INFO...!!!','warn','modal');
   pause(3)
   delete(hx)
   close(gcf)
   return
end
co = get(hObject,'color');
addpath(pwd,'database','codes')
```

```
if size(ls('database'),2) == 2
   delete('features.mat');
   delete('info.mat');
end
% Get default command line output from handles structure
varargout{1} = handles.output;



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if                  ispc                &&                isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
p  = get(handles.edit1,'UserData');
if strcmp(p,'123') == 1
   delete(hObject);
   delete(handles.pushbutton2)
   delete(handles.edit1);
   delete(handles.text2);
   delete(handles.text3);
```

```
    delete(handles.text1);
    delete(handles.text4);
    msgbox('WHY DONT U READ HELP BEFORE STARTING','HELP....!!!','help','modal')
    set(handles.AD_NW_IMAGE,'enable','on')
    set(handles.DE_LETE,'enable','on')
    set(handles.TRAIN_ING,'enable','on')
    set(handles.STA_RT,'enable','on')
    set(handles.RESET_ALL,'enable','on')
    set(handles.EXI_T,'enable','on')
    set(handles.HE_LP,'enable','on')
    set(handles.DATA_BASE,'enable','on')
    set(handles.text5,'visible','on')
else
    msgbox('INVALID PASSWORD FRIEND... XX','WARNING....!!!','warn','modal')
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close gcf

% --------------------------------------------------------------------
function AD_NW_IMAGE_Callback(hObject, eventdata, handles)
% hObject    handle to AD_NW_IMAGE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function DE_LETE_Callback(hObject, eventdata, handles)
% hObject    handle to DE_LETE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function TRAIN_ING_Callback(hObject, eventdata, handles)
% hObject    handle to TRAIN_ING (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% --------------------------------------------------------------------
function STA_RT_Callback(hObject, eventdata, handles)
% hObject    handle to STA_RT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function DATA_BASE_Callback(hObject, eventdata, handles)
% hObject    handle to DATA_BASE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function RESET_ALL_Callback(hObject, eventdata, handles)
% hObject    handle to RESET_ALL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function EXI_T_Callback(hObject, eventdata, handles)
% hObject    handle to EXI_T (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function HE_LP_Callback(hObject, eventdata, handles)
% hObject    handle to HE_LP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function READ_ME_Callback(hObject, eventdata, handles)
% hObject    handle to READ_ME (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
winopen('help.pdf')

% --------------------------------------------------------------------
function PRE_CAP_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to PRE_CAP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if exist('features.mat','file') == 0
   msgbox('FIRST TRAIN YOUR DATABASE','INFO...!!!','MODAL')
   return
end
ff = dir('database');
if length(ff) == 2
   h = waitbar(0,'Plz wait Matlab is scanning ur database...','name','SCANNING IS IN
PROGRESS');
   for k = 1:100
      waitbar(k/100)
      pause(0.03)
   end
   close(h)
   msgbox({'NO IMAGE FOUND IN DATABASE';'FIRST LOAD YOUR
DATABASE';'USE "ADD NEW IMAGE" MENU'},'WARNING....!!!','WARN','MODAL')
   return
end
fd = vision.CascadeObjectDetector();
[f,p] = uigetfile('*.jpg','PLEASE SELECT AN FACIAL IMAGE');
if f == 0
   return
end
p1 = fullfile(p,f);
im = imread(p1);
bbox = step(fd, im);
vo = insertObjectAnnotation(im,'rectangle',bbox,'FACE');
r = size(bbox,1);
if isempty(bbox)
   axes(handles.axes1)
   imshow(vo);
   msgbox({'NO FACE IN THIS PIC IS FOUND IN DB';'PLEASE SELECT SINGLE FACE
IMAGE'},'WARNING...!!!','warn','modal')
   uiwait
   cla(handles.axes1); reset(handles.axes1); set(handles.axes1,'box','on','xtick',[],'ytick',[])
   return
elseif r > 1
   axes(handles.axes1)
   imshow(vo);
   msgbox({'TOO MANY FACES IN THIS PIC';'PLEASE SELECT SINGLE FACE
IMAGE'},'WARNING...!!!','warn','modal')
   uiwait
```

```
    cla(handles.axes1); reset(handles.axes1); set(handles.axes1,'box','on','xtick',[],'ytick',[])
    return
end
axes(handles.axes1)
image(vo);
set(handles.axes1,'xtick',[],'ytick',[],'box','on')
bx = questdlg({'CORRECT IMAGE IS SELECTED';'SELECT OPTION FOR FACE EXTRACTION'},'SELECT AN OPTION','MANUALLY','AUTO','CC');
if strcmp(bx,'MANUALLY') == 1
    while 1
        fhx = figure(2);
        set(fhx,'menubar','none','numbertitle','off','name','PREVIEW')
        imc = imcrop(im);
        bbox1 = step(fd, imc);
        if size(bbox1,1) ~= 1
            msgbox({'YOU HAVENT CROPED A FACE';'CROP AGAIN'},'BAD ACTION','warn','modal')
            uiwait
        else
            close gcf
            break
        end
        close gcf
    end
 imc = imresize(imc,[300  300]);
 image(imc)
 text(20,20,'\bfUr Precaptured image.','fontsize',12,'color','y','fontname','comic sans ms')
 set(handles.axes1,'xtick',[],'ytick',[],'box','on')
end
 if strcmp(bx,'AUTO') == 1
    imc = imcrop(im,[bbox(1)-50  bbox(2)-250   bbox(3)+100    bbox(4)+400]);
    fhx = figure(2);
    set(fhx,'menubar','none','numbertitle','off','name','PREVIEW')
    imshow(imc)
    qx = questdlg({'ARE YOU SATISFIED WITH THE RESULTS?';' ';'IF YES THEN PROCEED';' ';'IF NOT BETTER DO MANUAL CROPING'},'SELECT','PROCEED','MANUAL','CC');
    if strcmpi(qx,'proceed') == 1
        close gcf
        imc = imresize(imc,[300  300]);
        axes(handles.axes1)
        image(imc)
        text(20,20,'\bfUr Precaptured image.','fontsize',12,'color','y','fontname','comic sans ms')
        set(handles.axes1,'xtick',[],'ytick',[],'box','on')
```

```
    elseif strcmpi(qx,'manual') == 1
        while 1
            fhx = figure(2);
            set(fhx,'menubar','none','numbertitle','off','name','PREVIEW')
            imc = imcrop(im);
            bbox1 = step(fd, imc);
            if size(bbox1,1) ~= 1
                msgbox({'YOU HAVENT CROPED A FACE';'CROP AGAIN'},'BAD
ACTION','warn','modal')
                uiwait
            else
                break
            end
            close gcf
        end
    close gcf
    imc = imresize(imc,[300 300]);
    axes(handles.axes1)
    image(imc)
    text(20,20,'\bfUr Precaptured image.','fontsize',12,'color','y','fontname','comic sans ms')
    set(handles.axes1,'xtick',[],'ytick',[],'box','on')
    else
    end
 end
 immxx = getimage(handles.axes1);
 zz = findsimilar(immxx);
 zz = strtrim(zz);
fxz = imread(['database/'  zz]);
q1= ehd(immxx,0.1);
q2 = ehd(fxz,0.1);
q3 = pdist([q1 ; q2]);
disp(q3)
if q3 < 0.5
    axes(handles.axes2)
    image(fxz)
    set(handles.axes1,'xtick',[],'ytick',[],'box','on')
    text(20,20,'\bfUr Database Entered Image.','fontsize',12,'color','y','fontname','comic sans
ms')
    set(handles.axes2,'xtick',[],'ytick',[],'box','on')
    xs = load('info.mat');
    xs1 = xs.z2;
    for k = 1:length(xs1)
        st = xs1{k};
        stx = st{1};
```

```
        if strcmp(stx,zz) == 1
            str = st{2};
            break
        end
    end
    fid = fopen('attendence_sheet.txt','a');
    fprintf(fid,'%s %s %s %s\r\n\n', 'Name','Date','Time', 'Attendence');
    c = clock;
    if c(4) > 12
        s = [num2str(c(4)-12) ,':',num2str(c(5)), ':', num2str(round(c(6))) ];
    else
        s = [num2str(c(4)) ,':',num2str(c(5)), ':', num2str(round(c(6))) ];
    end
    fprintf(fid,'%s %s %s %s\r\n\n', str, date,s,'Present');
    fclose(fid);
    set(handles.text5,'string',['Hello ' str ' ,Your attendence has been Marked.'])
    try
        s = serial('com22');
        fopen(s);
        fwrite(s,'A');
        pause(1)
        fclose(s);
        clear s
    catch
        msgbox({'PLZ CONNECT CABLE OR';'INVALID COM PORT
SELECTED'},'WARNING','WARN','MODAL')
        uiwait
        delete(s)
        clear s
    end
else
    msgbox('YOU ARE NOT A VALID PERSON', 'WARNING','WARN','MODAL')
    cla(handles.axes1)
    reset(handles.axes1)
    cla(handles.axes2)
    reset(handles.axes2)
    set(handles.axes1,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431 0.5176
0.7804],'linewidth',1.5);
    set(handles.axes2,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431 0.5176
0.7804],'linewidth',1.5)
end


% ------------------------------------------------------------------
```

```
function LIVE_CAM_Callback(hObject, eventdata, handles)
% hObject    handle to LIVE_CAM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global co
if exist('features.mat','file') == 0
   msgbox('FIRST TRAIN YOUR DATABASE','INFO...!!!','MODAL')
   return
end
ff = dir('database');
if length(ff) == 2
   h = waitbar(0,'Plz wait Matlab is scanning ur database...','name','SCANNING IS IN
PROGRESS');
   for k = 1:100
      waitbar(k/100)
      pause(0.03)
   end
   close(h)
   msgbox({'NO IMAGE FOUND IN DATABASE';'FIRST LOAD YOUR
DATABASE';'USE "ADD NEW IMAGE" MENU'},'WARNING....!!!','WARN','MODAL')
   return
end

if isfield(handles,'vdx')
   vid = handles.vdx;
   stoppreview(vid)
   delete(vid)
   handles = rmfield(handles,'vdx');
   guidata(hObject,handles)
   cla(handles.axes1)
   reset(handles.axes1)
   set(handles.axes1,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431    0.5176
0.7804],'linewidth',1.5)
   cla(handles.axes2)
   reset(handles.axes2)
   set(handles.axes2,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431    0.5176
0.7804],'linewidth',1.5)
end
info = imaqhwinfo('winvideo');
did = info.DeviceIDs;
if isempty(did)
   msgbox({'YOUR SYSTEM DO NOT HAVE A WEBCAM';' ';'CONNECT A
ONE'},'WARNING....!!!!','warn','modal')
   return
```

```
end
fd = vision.CascadeObjectDetector();
did = cell2mat(did);
for k = 1:length(did)
   devinfo = imaqhwinfo('winvideo',k);
   na(1,k) = {devinfo.DeviceName};
   sr(1,k) = {devinfo.SupportedFormats};
end
[a,b] = listdlg('promptstring','SELECT A WEB CAM DEVICE','liststring',na,'ListSize', [125,
75],'SelectionMode','single');
if b == 0
   return
end
if b ~= 0
   frmt = sr{1,a};
   [a1,b1] = listdlg('promptstring','SELECT RESOLUTION','liststring',frmt,'ListSize', [150,
100],'SelectionMode','single');
   if b1 == 0
      return
   end
end
frmt = frmt{a1};
l = find(frmt == '_');
res = frmt(l+1 : end);
l = find(res == 'x');
res1 = str2double(res(1: l-1));
res2 = str2double(res(l+1 : end));
axes(handles.axes1)
vid = videoinput('winvideo', a);
vr = [res1   res2];
nbands = get(vid,'NumberofBands');
h2im = image(zeros([vr(2) vr(1) nbands] , 'uint8'));
preview(vid,h2im);
handles.vdx = vid;
guidata(hObject,handles)
tx = msgbox('PLZ STAND IN FRONT OF CAMERA STILL','INFO.....!!!');
pause(1)
delete(tx)
kx = 0;
while 1
   im = getframe(handles.axes1);
   im = im.cdata;
   bbox = step(fd, im);
   vo = insertObjectAnnotation(im,'rectangle',bbox,'FACE');
```

```
    axes(handles.axes2)
    imshow(vo)
    if size(bbox,1) > 1
        msgbox({'TOO MANY FACES IN FRAME';' ';'ONLY ONE FACE IS
ACCEPTED'},'WARNING.....!!!','warn','modal')
        uiwait
        stoppreview(vid)
        delete(vid)
        handles = rmfield(handles,'vdx');
        guidata(hObject,handles)
        cla(handles.axes1)
        reset(handles.axes1)
        set(handles.axes1,'box','on','xtick',[],'ytick',[],'xcolor',[1 1 1],'ycolor',[1 1
1],'color',co,'linewidth',1.5)
        cla(handles.axes2)
        reset(handles.axes2)
        set(handles.axes2,'box','on','xtick',[],'ytick',[],'xcolor',[1 1 1],'ycolor',[1 1
1],'color',co,'linewidth',1.5)
        return
    end
    kx = kx + 1;
    if kx > 10 && ~isempty(bbox)
        break
    end
end
imc = imcrop(im,[bbox(1)+3 bbox(2)-35 bbox(3)-10 bbox(4)+70]);
imx = imresize(imc,[300 300]);
axes(handles.axes1)
image(imx)
text(20,20,'\bfUr Current image.','fontsize',12,'color','y','fontname','comic sans ms')
set(handles.axes1,'xtick',[],'ytick',[],'box','on')
 immxx = imx;
 zz = findsimilar(immxx);
 zz = strtrim(zz);
fxz = imread(['database/' zz]);
q1= ehd(immxx,0.1);
q2 = ehd(fxz,0.1);
q3 = pdist([q1 ; q2]);
disp(q3)
if q3 < 0.5
    axes(handles.axes2)
    image(fxz)
    set(handles.axes1,'xtick',[],'ytick',[],'box','on')
```

```
    text(20,20,'\bfUr Database Entered Image.','fontsize',12,'color','y','fontname','comic sans ms')
    set(handles.axes2,'xtick',[],'ytick',[],'box','on')
    xs = load('info.mat');
    xs1 = xs.z2;
    for k = 1:length(xs1)
        st = xs1{k};
        stx = st{1};
        if strcmp(stx,zz) == 1
            str = st{2};
            break
        end
    end
    fid = fopen('attendence_sheet.txt','a');
    fprintf(fid,'%s %s %s %s\r\n\n', 'Name','Date','Time', 'Attendance');
    c = clock;
    if c(4) > 12
        s = [num2str(c(4)-12) ,':',num2str(c(5)), ':', num2str(round(c(6))) ];
    else
        s = [num2str(c(4)) ,':',num2str(c(5)), ':', num2str(round(c(6))) ];
    end
    fprintf(fid,'%s %s %s %s\r\n\n', str, date,s,'Present');
    fclose(fid);
    set(handles.text5,'string',['Hello ' str ' ,Your attendence has been Marked.'])
    try
        s = serial('com22');
        fopen(s);
        fwrite(s,'A');
        pause(1)
        fclose(s);
        clear s
    catch
        msgbox({'PLZ CONNECT CABLE OR';'INVALID COM PORT SELECTED'},'WARNING','WARN','MODAL')
        uiwait
        delete(s)
        clear s
    end
else
    msgbox('YOU ARE NOT A VALID PERSON', 'WARNING','WARN','MODAL')
    cla(handles.axes1)
    reset(handles.axes1)
    cla(handles.axes2)
    reset(handles.axes2)
```

```
   set(handles.axes1,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431    0.5176
0.7804],'linewidth',1.5);
   set(handles.axes2,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431    0.5176
0.7804],'linewidth',1.5)
end
% --------------------------------------------------------------------
function SINGL_PIC_Callback(hObject, eventdata, handles)
% hObject    handle to SINGL_PIC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
flist  =  dir('database');
if length(flist) == 2
   msgbox('NOTHING TO DELETE','INFO','modal');
   return
end
cd('database')
[f,p] = uigetfile('*.jpg','SELECT A PIC TO DELETE IT');
if f == 0
   cd ..
   return
end
p1 = fullfile(p,f);
delete(p1)
flist  =  dir(pwd);
if length(flist) == 2
   cd ..
   return
end
for k =  3:length(flist)
   z = flist(k).name;
   z(strfind(z,'.') : end) = [];
   nlist(k-2) = str2double(z);
end
nlist  = sort(nlist);
h = waitbar(0,'PLZ WAIT, WHILE MATLAB IS RENAMING','name','PROGRESS...');
for k = 1:length(nlist)
   if k ~= nlist(k)
      p = nlist(k);
      movefile([num2str(p)  '.jpg']  ,  [num2str(k)  '.jpg'])
      waitbar((k-2)/length(flist),h,sprintf('RENAMED      %s      to      %s',[num2str(p)
'.jpg'],[num2str(k)  '.jpg']))
   end
   pause(.5)
end
```

```
close(h)
cd ..


% --------------------------------------------------------------------
function MULTI_PIC_Callback(hObject, eventdata, handles)
% hObject    handle to MULTI_PIC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
flist  =  dir('database');
if length(flist) == 2
   msgbox('NOTHING TO DELETE','INFO','modal');
   return
end
for k =  3:length(flist)
   na1(k-2,1) = {flist(k).name};
end
[a,b] = listdlg('promptstring','SELECT FILE/FILES TO DELETE','liststring',na1,'listsize',[125
100]);
if b == 0
   return
end
cd ('database')
for k = 1:length(a)
   str = na1{k};
   delete(str)
end
cd ..
flist  =  dir('database');
if length(flist) == 2
   msgbox({'NOTHING TO RENAME';'ALL DELETED'},'INFO','modal');
   return
end
cd('database')
flist  =  dir(pwd);
for k =  3:length(flist)
   z = flist(k).name;
   z(strfind(z,'.') : end) = [];
   nlist(k-2) = str2double(z);
end
nlist  = sort(nlist);
h = waitbar(0,'PLZ WAIT, WHILE MATLAB IS RENAMING','name','PROGRESS...');
for k = 1:length(nlist)
   if k ~= nlist(k)
```

```
        p = nlist(k);
        movefile([num2str(p)  '.jpg'] ,  [num2str(k)  '.jpg'])
        waitbar((k-2)/length(flist),h,sprintf('RENAMED %s to %s',[num2str(p)
'.jpg'],[num2str(k)  '.jpg']))
    end
    pause(.5)
end
close(h)
cd ..


% ---------------------------------------------------------------------
function BR_OWSE_Callback(hObject, eventdata, handles)
% hObject    handle to BR_OWSE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[f,p] = uigetfile('*.jpg','PLEASE SELECT AN FACIAL IMAGE');
if f == 0
    return
end
p1 = fullfile(p,f);
im = imread(p1);
fd = vision.CascadeObjectDetector();
bbox = step(fd, im);
vo = insertObjectAnnotation(im,'rectangle',bbox,'FACE');
r = size(bbox,1);
if isempty(bbox)
    fhx = figure(2);
    set(fhx,'menubar','none','numbertitle','off','name','PREVIEW')
    imshow(vo);
    msgbox({'WHAT HAVE U CHOOSEN?';'NO FACE FOUND IN THIS PIC,';'SELECT
SINGLE FACE IMAGE.'},'WARNING...!!!','warn','modal')
    uiwait
    delete(fhx)
    return
elseif r > 1
    fhx = figure(2);
    set(fhx,'menubar','none','numbertitle','off','name','PREVIEW')
    imshow(vo);
    msgbox({'TOO MANY FACES IN THIS PIC';'PLEASE SELECT SINGLE FACE
IMAGE'},'WARNING...!!!','warn','modal')
    uiwait
    delete(fhx)
    return
```

```
end
bx = questdlg({'CORRECT IMAGE IS SELECTED';'SELECT OPTION FOR FACE EXTRACTION'},'SELECT AN OPTION','MANUALLY','AUTO','CC');
if strcmp(bx,'MANUALLY') == 1
   while 1
      fhx = figure(2);
      set(fhx,'menubar','none','numbertitle','off','name','PREVIEW')
      imc = imcrop(im);
      bbox1 = step(fd, imc);
      if size(bbox1,1) ~= 1
         msgbox({'YOU HAVENT CROPED A FACE';'CROP AGAIN'},'BAD ACTION','warn','modal')
         uiwait
      else
         break
      end
      close gcf
   end
   close gcf
   imc = imresize(imc,[300  300]);
   cd ('database');
   l = length(dir(pwd));
   n = [int2str(l-1)    '.jpg'];
   imwrite(imc,n);
   cd ..
   while 1
      qq = inputdlg('WHAT IS UR NAME?','FILL');
      if isempty(qq)
         msgbox({'YOU HAVE TO ENTER A NAME';' ';'YOU CANT CLICK CANCEL'},'INFO','HELP','MODAL')
         uiwait
      else
         break
      end
   end
   qq = qq{1};
   if exist('info.mat','file') == 2
      load ('info.mat')
      r = size(z2,1);
      z2{r+1,1} = {n , qq};
      save('info.mat','z2')
   else
      z2{1,1} = {n,qq};
      save('info.mat','z2')
```

```
    end
end
if strcmp(bx,'AUTO') == 1
    imc = imcrop(im,[bbox(1)-50  bbox(2)-250   bbox(3)+100    bbox(4)+400]);
    fhx = figure(2);
    set(fhx,'menubar','none','numbertitle','off','name','PREVIEW')
    imshow(imc)
    qx = questdlg({'ARE YOU SATISFIED WITH THE RESULTS?';' ';'IF YES THEN PROCEED';' ';'IF NOT BETTER DO MANUAL CROPING'},'SELECT','PROCEED','MANUAL','CC');
    if strcmpi(qx,'proceed') == 1
        imc = imresize(imc,[300  300]);
        cd ('database');
        l = length(dir(pwd));
        n = [int2str(l-1)    '.jpg'];
        imwrite(imc,n);
        cd ..
        while 1
        qq = inputdlg('WHAT IS UR NAME?','FILL');
        if isempty(qq)
            msgbox({'YOU HAVE TO ENTER A NAME';' ';'YOU CANT CLICK CANCEL'},'INFO','HELP','MODAL')
            uiwait
        else
            break
        end
        end
        qq = qq{1};
    if exist('info.mat','file') == 2
        load ('info.mat')
        r = size(z2,1);
        z2{r+1,1} = {n , qq};
        save('info.mat','z2')
    else
        z2{1,1} = {n,qq};
        save('info.mat','z2')
    end
    close gcf
    elseif strcmpi(qx,'manual') == 1
        while 1
            fhx = figure(2);
            set(fhx,'menubar','none','numbertitle','off','name','PREVIEW')
            imc = imcrop(im);
            bbox1 = step(fd, imc);
```

```
        if size(bbox1,1) ~= 1
            msgbox({'YOU HAVENT CROPED A FACE';'CROP AGAIN'},'BAD
ACTION','warn','modal')
            uiwait
        else
            break
        end
        close gcf
    end
  close gcf
  imc = imresize(imc,[300  300]);
  cd ('database');
  l = length(dir(pwd));
  n = [int2str(l-1)    '.jpg'];
  imwrite(imc,n);
  cd ..
  while 1
    qq = inputdlg('WHAT IS UR NAME?','FILL');
    if isempty(qq)
        msgbox({'YOU HAVE TO ENTER A NAME';' ';'YOU CANT CLICK
CANCEL'},'INFO','HELP','MODAL')
        uiwait
    else
        break
    end
  end
  qq = qq{1};
  if exist('info.mat','file') == 2
    load ('info.mat')
    r = size(z2,1);
    z2{r+1,1} = {n , qq};
    save('info.mat','z2')
  else
    z2{1,1} = {n,qq};
    save('info.mat','z2')
  end
  else
    return
  end
end


% --------------------------------------------------------------------
function FRM_CAM_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to FRM_CAM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global co
if isfield(handles,'vdx')
    vid = handles.vdx;
    stoppreview(vid)
    delete(vid)
    handles = rmfield(handles,'vdx');
    guidata(hObject,handles)
    cla(handles.axes1)
    reset(handles.axes1)
    set(handles.axes1,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431     0.5176
0.7804],'linewidth',1.5)
    cla(handles.axes2)
    reset(handles.axes2)
    set(handles.axes2,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431     0.5176
0.7804],'linewidth',1.5)
end
fd = vision.CascadeObjectDetector();
info = imaqhwinfo('winvideo');
did = info.DeviceIDs;
if isempty(did)
    msgbox({'YOUR SYSTEM DO NOT HAVE A WEBCAM';' ';'CONNECT A
ONE'},'WARNING....!!!!','warn','modal')
    return
end
did = cell2mat(did);
for k = 1:length(did)
    devinfo = imaqhwinfo('winvideo',k);
    na(1,k) = {devinfo.DeviceName};
    sr(1,k) = {devinfo.SupportedFormats};
end
[a,b] = listdlg('promptstring','SELECT A WEB CAM DEVICE','liststring',na,'ListSize', [125,
75],'SelectionMode','single');
if b == 0
    return
end
if b ~= 0
    frmt = sr{1,a};
    [a1,b1] = listdlg('promptstring','SELECT RESOLUTION','liststring',frmt,'ListSize', [150,
100],'SelectionMode','single');
    if b1 == 0
        return
```

```
    end
end
frmt = frmt{a1};
l = find(frmt == '_');
res = frmt(l+1 : end);
l = find(res == 'x');
res1 = str2double(res(1: l-1));
res2 = str2double(res(l+1 : end));
axes(handles.axes1)
vid = videoinput('winvideo', a);
vr  =  [res1    res2];
nbands  =  get(vid,'NumberofBands');
h2im  = image(zeros([vr(2)  vr(1)  nbands] , 'uint8'));
preview(vid,h2im);
handles.vdx = vid;
guidata(hObject,handles)
tx = msgbox('PLZ STAND IN FRONT OF CAMERA STILL','INFO......!!!');
pause(1)
delete(tx)
kx  =  0;
while 1
    im = getframe(handles.axes1);
    im =  im.cdata;
    bbox = step(fd, im);
    vo = insertObjectAnnotation(im,'rectangle',bbox,'FACE');
    axes(handles.axes2)
    imshow(vo)
    if size(bbox,1) > 1
        msgbox({'TOO MANY FACES IN FRAME';' ';'ONLY ONE FACE IS
ACCEPTED'},'WARNING.....!!!','warn','modal')
        uiwait
        stoppreview(vid)
        delete(vid)
        handles = rmfield(handles,'vdx');
        guidata(hObject,handles)
        cla(handles.axes1)
        reset(handles.axes1)
        set(handles.axes1,'box','on','xtick',[],'ytick',[],'xcolor',[1          1          1],'ycolor',[1          1
1],'color',co,'linewidth',1.5)
        cla(handles.axes2)
        reset(handles.axes2)
        set(handles.axes2,'box','on','xtick',[],'ytick',[],'xcolor',[1          1          1],'ycolor',[1          1
1],'color',co,'linewidth',1.5)
        return
```

```
    end
    kx = kx + 1;
    if kx > 10 && ~isempty(bbox)
        break
    end
end
imc = imcrop(im,[bbox(1)+3    bbox(2)-35    bbox(3)-10      bbox(4)+70]);
imx  =   imresize(imc,[300  300]);
fhx = figure(2);
set(fhx,'menubar','none','numbertitle','off','name','PREVIEW')
imshow(imx)
cd ('database');
l = length(dir(pwd));
n = [int2str(l-1)    '.jpg'];
imwrite(imx,n);
cd ..
    while 1
        qq = inputdlg('WHAT IS YOUR NAME?','FILL');
        if isempty(qq)
            msgbox({'YOU HAVE TO ENTER A NAME';' ';'YOU CANT CLICK
CANCEL'},'INFO','HELP','MODAL')
            uiwait
        else
            break
        end
    end
    qq = qq{1};
    if exist('info.mat','file') == 2
        load ('info.mat')
        r = size(z2,1);
        z2{r+1,1} = {n , qq};
        save('info.mat','z2')
    else
        z2{1,1} = {n,qq};
        save('info.mat','z2')
    end
close gcf
stoppreview(vid)
delete(vid)
handles = rmfield(handles,'vdx');
guidata(hObject,handles)
cla(handles.axes1)
reset(handles.axes1)
```

```
set(handles.axes1,'box','on','xtick',[],'ytick',[],'xcolor',[1 1 1],'ycolor',[1 1 1],'color',co,'linewidth',1.5)
cla(handles.axes2)
reset(handles.axes2)
set(handles.axes2,'box','on','xtick',[],'ytick',[],'xcolor',[1 1 1],'ycolor',[1 1 1],'color',co,'linewidth',1.5)


% --- Executes on key press with focus on edit1 and none of its controls.
function edit1_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  structure with the following fields (see UICONTROL)
%        Key: name of the key that was pressed, in lower case
%        Character: character interpretation of the key(s) that was pressed
%        Modifier: name(s) of the modifier key(s) (i.e., control, shift) pressed
% handles    structure with handles and user data (see GUIDATA)
pass = get(handles.edit1,'UserData');
v = double(get(handles.figure1,'CurrentCharacter'));
if v == 8
   pass = pass(1:end-1);
   set(handles.edit1,'string',pass)
elseif any(v == 65:90) || any(v  == 97:122) || any(v == 48:57)
   pass = [pass  char(v)];
elseif v == 13
   p  = get(handles.edit1,'UserData');
   if strcmp(p,'123') == true
      delete(hObject);
      delete(handles.pushbutton2)
      delete(handles.pushbutton1);
      delete(handles.text2);
      delete(handles.text3);
      delete(handles.text1);
      delete(handles.text4);
      msgbox('WHY DONT U READ HELP BEFORE STARTING','HELP....!!!','help','modal')
      set(handles.AD_NW_IMAGE,'enable','on')
      set(handles.DE_LETE,'enable','on')
      set(handles.TRAIN_ING,'enable','on')
      set(handles.STA_RT,'enable','on')
      set(handles.RESET_ALL,'enable','on')
      set(handles.EXI_T,'enable','on')
      set(handles.HE_LP,'enable','on')
      set(handles.DATA_BASE,'enable','on')
      set(handles.text5,'visible','on')
```

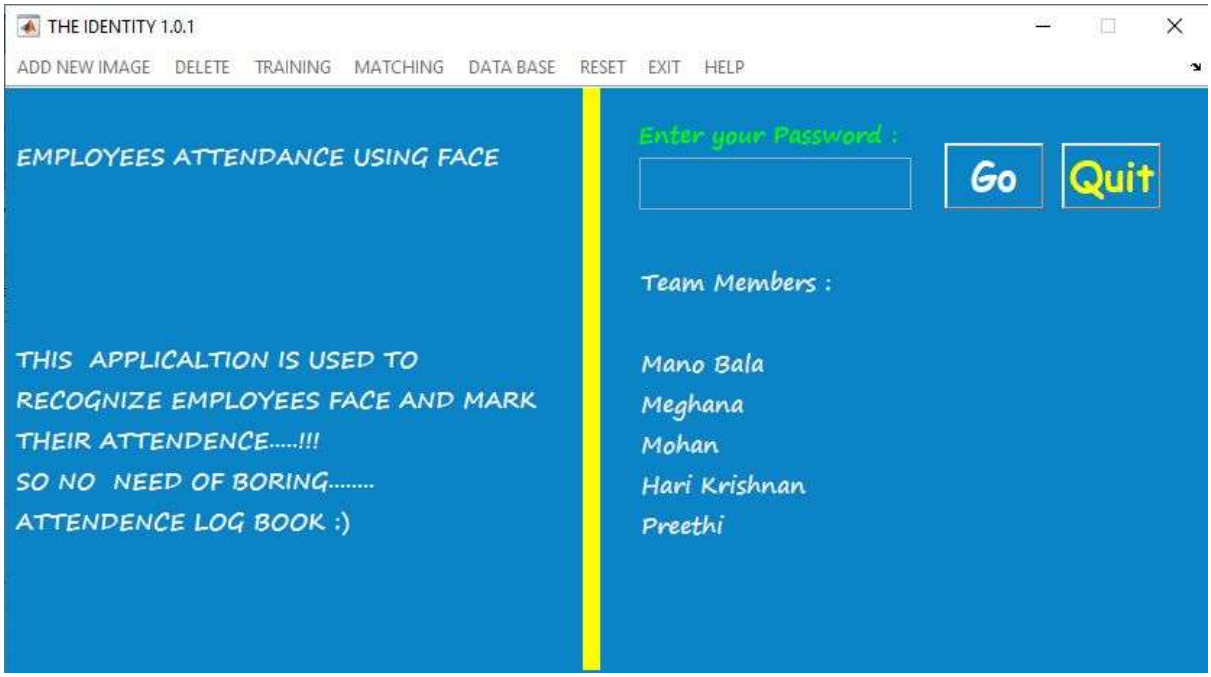
```
        return
    else
        beep
        msgbox('INVALID PASSWORD FRIEND... XX','WARNING....!!!','warn','modal')
        uiwait;
        set(handles.edit1,'string','')
        return
    end
else
    msgbox({'Invalid Password Character';'Can''t use Special Character'},'warn','modal')
    uiwait;
    set(handles.edit1,'string','')
    return
end
set(handles.edit1,'UserData',pass)
set(handles.edit1,'String',char('*'*sign(pass)))


% --------------------------------------------------------------------
function VI_EW_Callback(hObject, eventdata, handles)
% hObject    handle to VI_EW (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

f = dir('database');
if length(f) == 2
    msgbox('YOUR DATA BASE HAS NO IMAGE TO DISPLAY','SORRY','modal')
    return
end
l = length(f)-2;
while 1
    a = factor(l);
    if length(a) >= 4
        break
    end
    l = l+1;
end
d  = a(1: ceil(length(a)/2));
d = prod(d);
d1 = a(ceil(length(a)/2)+1  : end);
d1 = prod(d1);
zx = sort([d d1]);
figure('menubar','none','numbertitle','off','name','Images  of  Database','color',[0.0431    0.5176
0.7804],'position',[300 200 600 500])
```

```
for k = 3:length(f)
   im = imread(f(k).name);
   subplot(zx(1),zx(2),k-2)
   imshow(im)
   title(f(k).name,'fontsize',10,'color','w')
end



% --------------------------------------------------------------------
function Start_Training_Callback(hObject, eventdata, handles)
% hObject    handle to Start_Training (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ff = dir('database');
if length(ff) == 2
   h = waitbar(0,'Plz wait Matlab is scanning ur database...','name','SCANNING IS IN
PROGRESS');
   for k = 1:100
      waitbar(k/100)
      pause(0.03)
   end
   close(h)
   msgbox({'NO IMAGE FOUND IN DATABASE';'FIRST LOAD YOUR
DATABASE';'USE "ADD NEW IMAGE" MENU'},'WARNING....!!!','WARN','MODAL')
   return
end
if exist('features.mat','file') == 2
   bx = questdlg({'TRAINING HAS ALREDY BEEN DONE';' ';'WANT TO TRAIN
DATABASE AGAIN?'},'SELECT','YES','NO','CC');
   if strcmpi(bx,'yes') == 1
      builddatabase
      msgbox('TRAINING DONE....PRESS OK TO CONTINUE','OK','modal')
      return
   else
      return
   end
else
   builddatabase
   msgbox('TRAINING DONE....PRESS OK TO CONTINUE','OK','modal')
   return
end
```

```
% --------------------------------------------------------------------
function BYE_Callback(hObject, eventdata, handles)
% hObject    handle to BYE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close gcf


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%end%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% --------------------------------------------------------------------
function ATTENDENCE_Callback(hObject, eventdata, handles)
% hObject    handle to ATTENDENCE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if  exist('attendence_sheet.txt','file') == 2
    winopen('attendence_sheet.txt')
else
    msgbox('NO ATTENDENCE SHEET TO DISPLAY','INFO...!!!','HELP','MODAL')
end

% --------------------------------------------------------------------
function DEL_ATTENDENCE_Callback(hObject, eventdata, handles)
% hObject    handle to DEL_ATTENDENCE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if  exist('attendence_sheet.txt','file') == 2
   delete('attendence_sheet.txt')
   msgbox('ATTENDENCE DELETED','INFO...!!!','MODAL')
else
    msgbox('NO ATTENDENCE SHEET TO DELETE','INFO...!!!','HELP','MODAL')
end


% --------------------------------------------------------------------
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
x   = questdlg({'Resetting will Clear the followings: ';'1. Attendence_sheet';'2. Database';'3. features.mat';'4. Info.mat';'Do u want to continue?'},'Please select...!!');
if strcmpi(x,'yes') == 1
```

```
    delete('attendence_sheet.txt')
    delete('features.mat')
    delete('info.mat')
    cd ([pwd, '\database'])
    f = dir(pwd);
    for k = 1:length(f)
        delete(f(k).name)
    end
    cd ..
    cla(handles.axes1);
    reset(handles.axes1);
    set(handles.axes1,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431     0.5176
0.7804],'linewidth',1.5)
    cla(handles.axes2);
    reset(handles.axes2);
    set(handles.axes2,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431     0.5176
0.7804],'linewidth',1.5)
    set(handles.text5,'string','')
    beep
    msgbox('All Reset','Info','modal')
end


% --------------------------------------------------------------------
function Untitled_2_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
cla(handles.axes1);
reset(handles.axes1);
set(handles.axes1,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431       0.5176
0.7804],'linewidth',1.5)
cla(handles.axes2);
reset(handles.axes2);
set(handles.axes2,'box','on','xcolor','w','ycolor','w','xtick',[],'ytick',[],'color',[0.0431       0.5176
0.7804],'linewidth',1.5)
set(handles.text5,'string','')


% --------------------------------------------------------------------
function Untitled_3_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

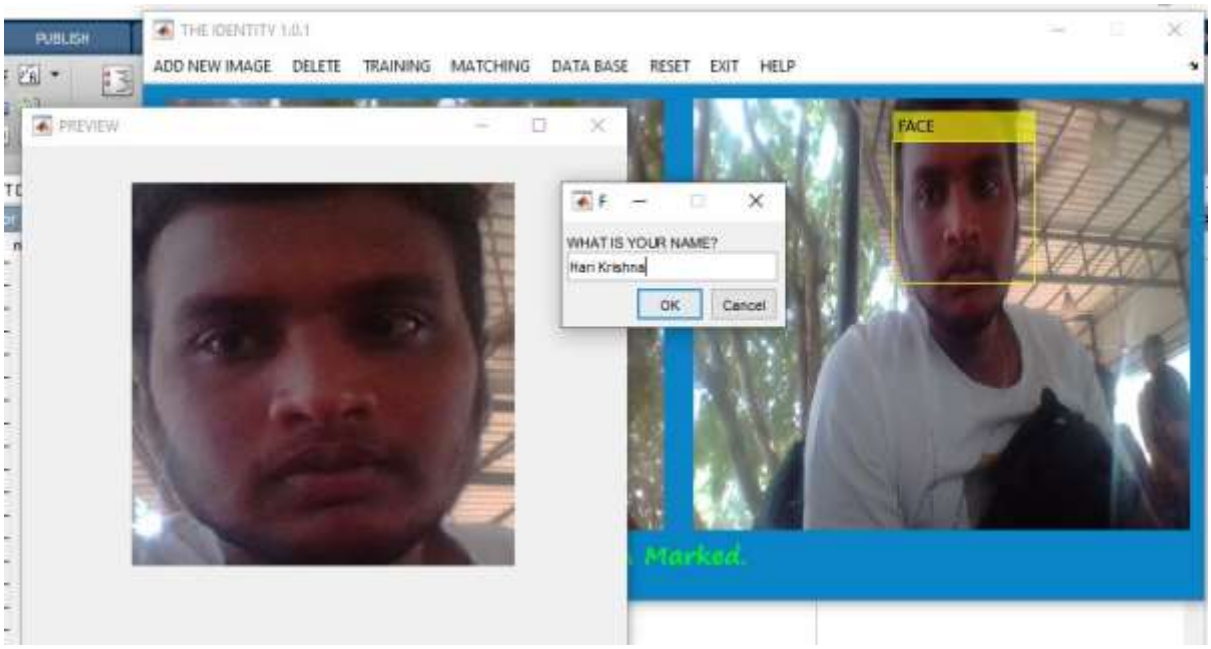
```
% --------------------------------------------------------------------
function Untitled_4_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --------------------------------------------------------------------
function Untitled_5_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```



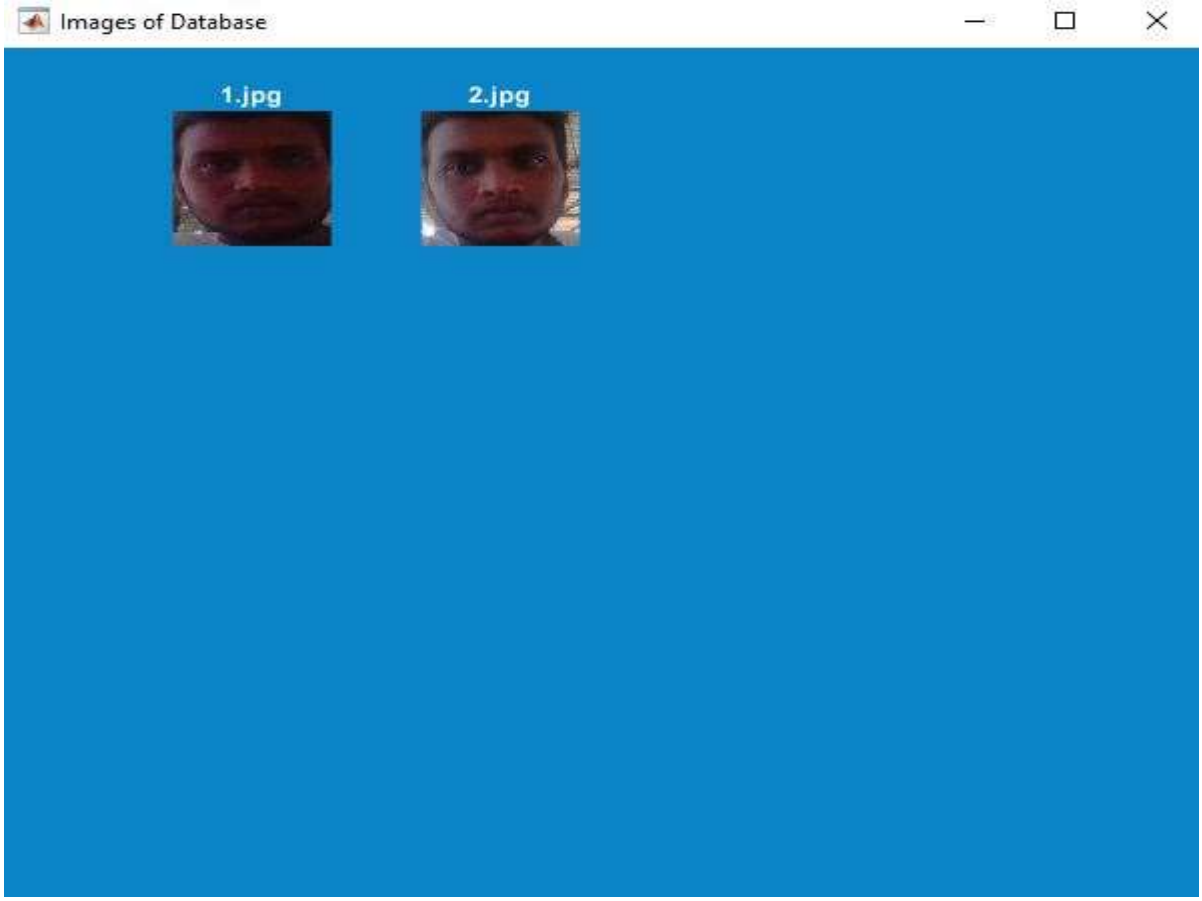
1. IMAGE ACQUISITION [TRAINING]

2. PRE PROCESSING

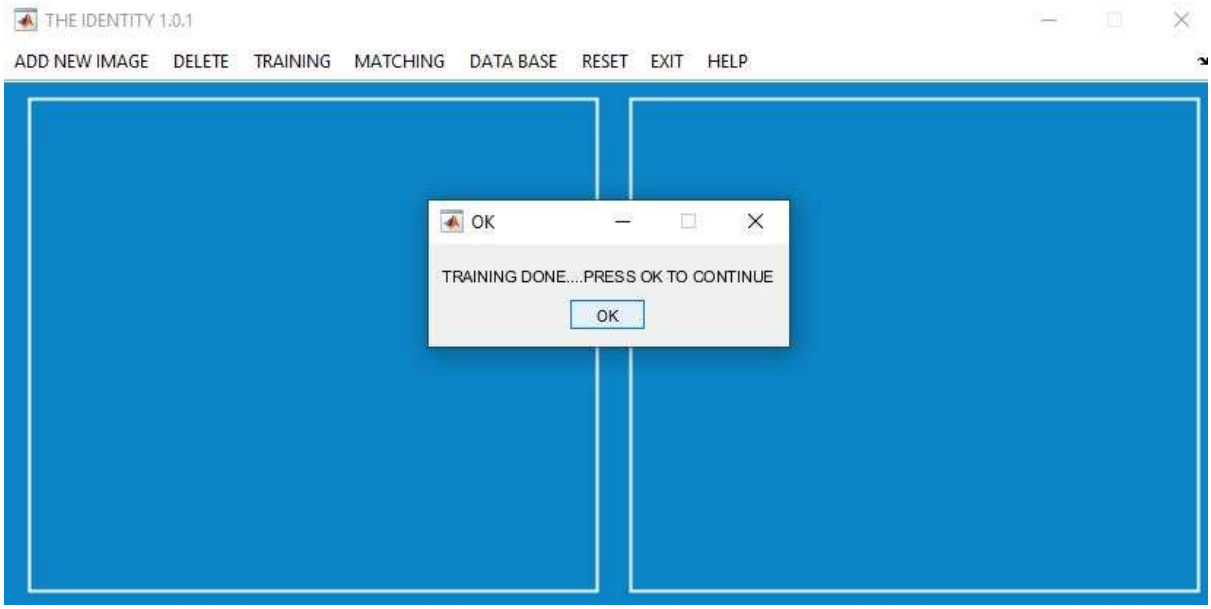
3. ENROLLMENT



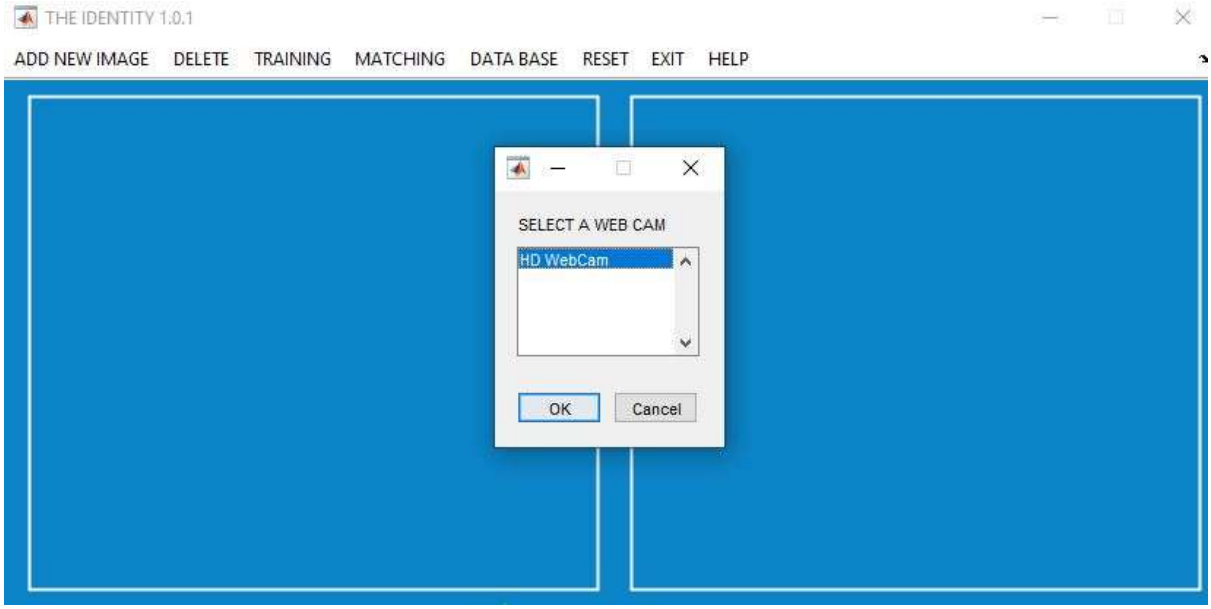
4. DATABASE



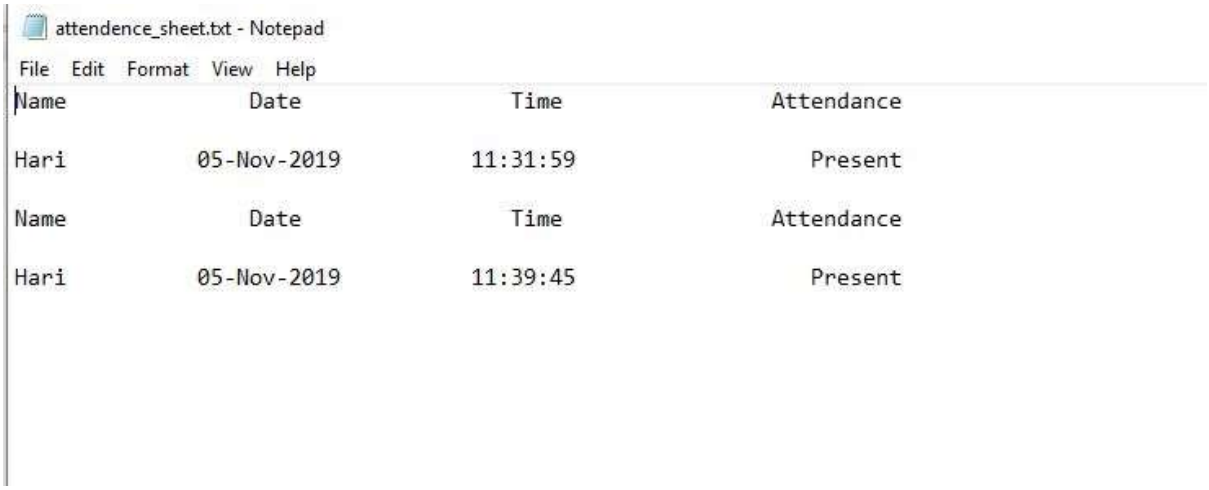
5. DETECTION



6. RECOGNITION



7. DISPLAY NAME/PRESENT/ABSENT


attendence_sheet.txt - Notepad

File Edit Format View Help

Name	Date	Time	Attendance
Hari	05-Nov-2019	11:31:59	Present
Name	Date	Time	Attendance
Hari	05-Nov-2019	11:39:45	Present

INVENTION DETAILS

Objects of the Invention:

It is very useful in protected zone while many pupils are entering and exiting.

For e.g., Take a bank while entering it takes the data from the user and the

cameras installed will takes the photos particulary the faces of the user and

combines with their user names.

Summary of Invention:

We use open CV with python .It detect the faces and combines with the ID

given by names and taken into database .Afterwards while recognizing the face it display the name.

FACE RECOGNITION SYSTEM:-

A **facial recognition system** is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source. There are multiple methods in which facial recognition systems work, but in general, they work by comparing selected facial features from given image with faces within a database. It is also described as a Biometric Artificial Intelligence based application that can uniquely identify a person by analysing patterns based on the person's facial textures and shape.

Detailed Description of the Invention:

1. We use MAT LAB and this detects the faces and store in the database our duty is to give the ID's for the corresponding image and the names.
2. While it about to sign in process it checks the image in the database whether it is present or not and gives.
3. The result our name with our image if not it displays "Not an authorized user please sign

REFERENCES:-

1. M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591. 1991.
2. Y. Zhang and C. Liu, "Face recognition using kemel principal component analysis and genetic algorithms," IEEE Workshop on Neural Networks for Signal Processing, pp. 4-6 Sept. 2002.
3. J. Zhu and Y. L. Yu, "Face Recognition with Eigenfaces," IEEE International Conference on Industrial Technology, pp. 434 -438, Dec. 1994.
4. M. H. Yang, N. Ahuja, and D. Kriegmao, "Face recognition using kernel eigenfaces," IEEE International Conference on Image Processing, vol. 1, pp. 10-13, Sept. 2000.
5. T. D. Russ, M. W. Koch, and C. Q. Little, "3D Facial Recognition: A Quantitative Analysis," 38th Annual 2004 International Carnahan Conference on Security Technology, 2004.
6. P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About," in Proceedings of the IEEE, vol. 94, Issue 11, 2006.
7. Y.-W. Kao, H.-Z. Gu, and S.-M. Yuan "Personal based authentication by face recognition," in proc. Fourth International Conference on Networked Computing and Advanced Information Management, pp 81-85, 2008.
8. A. T. Acharya and A. Ray, Image Processing: Principles and Applications, New York: Wiley, 2005
9. K. Cheng, L. Xiang, T. Hirota and K. Ushijima, "Effective Teaching for Large Classes with Rental PCs by Web System WTS," in Proc. Data Engineer- ing Workshop 2005 (DEWS2005), 2005, 1D-d3 (in Japanese).

10. W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosen- feld, “Face recognition: A literature survey,” ACM Computing Surveys, 2003, vol. 35, no. 4, pp. 399-458.

11. S. Nishiguchi, K. Higashi, Y. Kameda and M. Minoh, “A Sensor-fusion Method of Detecting A Speaking Student,” IEEE International Conference on Multi- media and Expo (ICME2003), 2003, vol. 2, pp. 677- 680.

12. R.E. Burkard and E. C ela, “Linear Assignment Prob- lems and Extensions”, In Handbook of Combinato- rial Optimization, Du Z, Pardalos P (eds). Kluwer Academic Publishers: Dordreck, 1999, pp. 75-149.

13. Taiping Zhang, Yuan Yan Tang, Bin Fang, Zhaowei Shang, Xiaoyu L

14. iu. Face Recognition Under Varying Illumination Using Gradient faces. In Image Processing, IEEE Transactions, Nov. 2009..

15. JA.C. Weaver, “Biometric authentication”, Computer, 39(2), pp 96-97 (2006).

16. J. Ortega-Garcia, J. Bigun, D. Reynolds and J.Gonzalez- Rodriguez, “Authentication gets personal with biometrics”, Signal Processing Magazine, IEEE, 21(2), pp 50-62 (2004).

17. Anil K. Jain, Arun Ross and Salil Prabhakar,” An introduction to biometric recognition”, Circuits and Systems for Video Technology, IEEE Transactions on Volume 14, Issue 1, Jan. 2004 Page(s):4 – 20.

18. Rowley, H. A., Baluja, S. and Kanade, T. (1998). "Neural network-based face detection." Pattern Analysis and Machine Intelligence, IEEE Transactions.

19. Osuna, E., Freund, R. and Girosit, F. (1997). "Training support vector machines: an application to face detection." 130-136.

20. Schneiderman, H. and T. Kanade (1998). Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp 45-51.

21. Viola, P. and M. J. Jones (2004). "Robust Real-Time Face Detection." International Journal of Computer Vision 57(2): 137-154.

22. Viola, P. and M. Jones (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition.

23. Shih, P. and C. Liu (2004). Face Detection Using Discriminating Feature Analysis and Support Vector Machine in Video, in: Proc. 17th Intl. Conf. Pattern Recognition, IEEE Computer Society Washington, DC, USA.

24. Curran, K., X. Li, et al. (2005). "Neural network face detection." Imaging Science Journal, the 53(2): 105-115.

25. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):711–720, July 1997.

----------END----------