NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade.

STUDENT ATTENDNACE MONITIORING SYSTEM (FACE RECOGNITION)

A MINI PROJECT

REPORT

Submitted by:

ADITYA JAKKARADDI	(1NH18EC700)
MANOJ N	(1NH18EC730)
RAJEEV KUMAR	(1NH18EC741)
S SHARAN RAO	(1NH18EC742)

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the mini project work entitled “**STUDENT ATTENDNACE MONITORING SYSTEM (FACE RECOGNITION)**” carried out by **ADITYA JAKKARADDI (1NH18EC700), MANOJ N (1NH18EC730), RAJEEV KUMAR (1NH18EC741), S SHARAN RAO(1NH18EC742),** bonafide students of Electronics and Communication Department, New Horizon College of Engineering, Bangalore.

The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

Project Guide	**HOD ECE**
- - - - - - - - - - - - - - - - -	- - - - - - - - - - - - - - - - -
Dr K.C.R NISHA	**DR. SANJEEV SHARMA**
Professor	Professor and HoD
Dept. of ECE	Dept. of ECE
NHCE	NHCE

<u>External Viva</u>

Name of Examiner	Signature with Date
1.	
2.	

ACKNOWLEDGEMENT

The satisfaction that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement helped us succeed.

We thank **Dr. Mohan Manghnani**, Chairman of **New Horizon Educational Institution**, for providing necessary infrastructure and creating good environment.

We also record here the constant encouragement and facilities extended to us by

Dr. Manjunatha, Principal, NHCE and **Dr. Sanjeev Sharma**, head of the department of Electronics and Communication Engineering. We extend sincere gratitude to them.

We sincerely acknowledge the encouragement, timely help and guidance to us by our beloved guide **DR K.C.R NISHA** to complete the project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of electronics and communication department for their co-operation extended to us, who helped us directly or indirectly in this successful completion of mini project.

ADITYA JAKKARADDI	**1NH18EC700**
MANOJ N	**1NH18EC730**
RAJEEV KUMAR	**1NH18EC741**
S SHARAN RAO	**1NH18EC742**

Abstract:

We are proposing a student attendance monitoring system which can be used in every organization to mark the attendance of students or employees. The main application of Student attendance monitoring system is seen in teaching institutions, where the attendance of students has to be regularly monitored on hourly basis. The method developed provides a secure and effective may recording attendance. Student attendance monitoring system uses mainly two algorithms i.e. Viola Jones Algorithm and Local binary pattern algorithm. Viola Jones algorithm is used for face detection and Local binary pattern is used for the feature extraction and face recognition. For classification support vector machine algorithm is used.

Initially every student must enroll his/her facial data into the system. When a student stands in front of the camera his/her face will be captured by the camera. This image will be processed by an enhanced face recognition algorithm written in MATLAB. The captured image

Will be verified against all the data in the file system which is linked to a database where information of every student will be stored.

Once the system verifies the image processed it will display the information of the student. This is will be linked to another system which will keep track of the student's attendance and will update the attendance accordingly.

TABLE OF CONTENTS

List of figures

CHAPTER - 1

INTRODUCTION

Face is one of the most distinctive feature of a human being. We tend to recognise a person by his or her face in a more convenient manner. We remember the faces better and that helps in building better connections. No two human beings have same facial features like eyes, nose, lips, ears, lips. When we connect all these features of a person we can actually distinguish the person from the others.

Since the face is such a distinct feature of a 1human being, and it is used to remember other humans. This face recognition is also being used in multiple industries as form of marking their presence in the premises. In other words, the industries today are resorting to different techniques in order to automate the process of marking someone's presence by using ways like Biometrics recognition, Iris recognition, Face recognition, etc.

Since the companies employ a large number of people, resorting to manual methods of marking the attendance is one of the most cumbersome process one has to undergo. With the advancement in the technologies and new systems coming into the market for the same, the companies started using these ways of marking attendance as these processes aren't cumbersome at all and also the process saves a lot of time for everyone. Using automation has been proved to be more efficient in most of the processes and the case is no different either.

The systems mentioned above work according to the following ways :

- Biometrics recognition : This software first scams the finger impression of the person on a scanner and store the information in the database. When the person places his or her finger on the scanner the next time, it compares the prints with the ones present in the database and then marks the attendance.
- Iris recognition : This software scans the eye of the person and stores the information in the database and when the person brings the eye near the scanner,

it compares the images with the ones present in the database and marks the attendance.

- Face recognition : This software records the images of a person using a camera and store them in the database. When the person once again comes in front of the camera then it compares the image with the ones in the database and marks the attendance.

1.1 Facial Recognition:

A facial recognition system is a technology used for identifying a person's face. It does so by, comparing the images captured by a device with a camera with the images stored in the database. It has a wide range of applications. Nowadays, the most common use is as the access control in the security systems in the corporate world. The advancement in the way this system evolved has made it possible to use for artificial imaging and has opened a lot of new avenues in the digital imaging world.

Fig. 1.1 Image of facial recognition at the street

The proposed project mainly focuses on the facial recognition system in use in order to solve the menace of a process that is taking attendance especially in educational institutions.

1.1.1 History of Facial Recognition:

The three main architects involved in the process of the development of the facial recognition system are:

- Woody Bledsoe
- Helen Chan Wolf
- Charles Bisson

The work for this kind of a system began in the years 1964-65 with Bledsoe being the primary scientist. The initial round of funding was provided by an unnamed intelligence agency and hence, a lot of the work was not published or known to the world.

The initial system used to map the face and its features like the distance between the eyes and nose and ears. Then it would be rotated by the computer in order to map and anticipate the distance between the features in case there was a variation in the expression or pose of the person. But selecting the similar range of photos from a wide range of images in the database was getting difficult.

Fig. 1.2 Cameras with the hardware of the facial recognition software

Then they started to maintain the distance of the face map manually. This process was called the man-machine as they would collect and feed the data of the distance and coordinates of the facial features manually and the image as well. With the advancement in technologies and the kind of up-gradation that happened in the processors and their speeds, they were able to fully automate the working of the system by segregating the data in a systematic manner so that the processor could recognise the face properly.

The major breakthrough came in the year 1997. That year, the students of University of Bochum and University of Southern California along with Professor Christoph von der Malsburg built the system which was the most accurate and efficient of its time. It was names as ZN- FACE. And was used by companies like the Deutsche bank and for the operations at the airports and other busy locations. Nowadays, 3-D face scans, high resolution face images and iris images have been put into use by highly complicated yet great machinery. The cameras and the software work in tandem and have reduced the error rate of the automatic face recognition systems by almost a factor of 272.

Fig. 1.3 Artificial impression of a face by a software

1.1.2 Techniques of Facial Recognition:

The current facial recognition systems employ the following techniques :

- <u>Traditional *:*</u> This technique involves face mapping discussed before. The system measures the distance between eyes, ears, nose lips and other physical features like the forehead to the bottom of the chin, etc. Then the data is stored and when the person comes in front of the camera the image is taken and the distance of the physical features is calculated and compared to the images and the data present in the database.

- <u>3-Dimensional Recognition *:*</u> This technique involves a much more in-depth study of the facial features of the person's image. The depth and the distance of the facial features is calculated and stored in the system by creating a 3 - dimensional face in the system by capturing the details of the face of the person. It uses symmetry and geometrical tools to create the details of the image and then process it. When the person comes in front of the camera, the camera captures the image and creates the artificial 3-dimensional image and compares the features of the face with the ones in the database. The depth of the eye sockets, the way the nose is aligned with the face and different points which help in determining the build of the face even if there is a variation in the pose, tilt or direction of the face while capturing are some of the things compared.

Fig. 1.4 Artificial impression of a face by a software

- <u>Skin Texture Analysis *:*</u> This technique involves the study of the surface of the skin. The system gathers information about the marks on the skin, unique patterns and line on the skin of the face of the person and then convert it into a mathematical space. This mathematical space is then used to compare the image captured by the camera with the data in the server. This is one of the most recently developed techniques and still not so widely used across the world.

- Facial Recognition combining different techniques *:* Here, the techniques mentioned above are used together in order to attain highest attainable accuracy and to speed up the process of facial recognition. Since there are advantages as well as disadvantages of all the systems, this kind of usage helps in covering up the disadvantages and also to help build it in a much more customisable manner according to the needs of the client. This helps in reducing the occurrence of errors to a great extent.

Fig. 1.5 Artistic representation of the process of comparing

- Thermal Cameras *:* This technique involves the use of thermal cameras. These cameras usually ignore the other parts of the subject and only focus on the mapping of the face of the subject. They even ignore the glasses or earrings and stuff and can store the entire shape of the head. These cameras use low-sensitive, low-resolution ferroelectric sensors which help in generating Long-Wave Thermal Infrared (LWIR).

The issue with this technique is that the database available is very limited and hence, the researchers are resorting to the data collected by the thermal as well as conventional cameras. The most important advantage of combining both the data from the thermal as well as conventional cameras is that the accuracy is almost 98.4% when used outdoors and 89% when used indoors.

CHAPTER - 2

LITERATURE REVIEW

2.1 PAPER 1:

PVN & Gupta, (2013) in their work titled “Fingerprint based student attendance system using GSM” conducted a student attendance management system using a GSM network by collecting the data on a finger print (biometric) scanner. Then when the student his or her finger on the biometric scanner, it compared the finger print with the ones already present in the database. It recorded the attendance and sent a message to the student's parents regarding his or her attendance report using the GSM - network of the mobile phone.

The limitation of this work is that it becomes very time consuming and a cumbersome process as each and every student has to come and wait for his or her turn in a queue in order to give the attendance.

2.2 PAPER 2:

Baban,(2014) in their work titled “Attendance checking system using Quick response code (QR)” made a student attendance management system to record, view, monitor and report a student's attendance by scanning a QR code using the scanners available on the smart phones.

The limitation of this work is that it is highly un-reliable as the quick response code can be duplicated pretty easily and the attendance can be recorded even if the student is not physically present in the classroom.

2.3 PAPER-3:

Hui et al, (2014) in their work titled “Mobile based attendance monitoring system” designed and developed a student attendance management system which uses the GPS software i order to locate the geographical location of the mobile device in order to mark the attendance of the student.

The limitation of this work is that if the smartphones are swapped with someone else or handed over to someone else then this may lead to proxy of the student. Hence, it is not a reliable system for recording attendance of students.

CHAPTER – 3

EXISTING SYSTEMS

This section presents the systems which already exist in the society and are also used by many but these systems need improvement. These systems need to be changed.

The two major existing systems most common in use are the following two :

- Fingerprint based recognition system
- Iris based recognition system

3.1 Finger Print Based Recognition System:

Each and every human being's finger prints are different. The impressions are different on the fingers of the same hand as well. TFhe fingerprint is one of the most important of identifying a person and is collected by most government organizations while issuing official documents.

This system consists of a database which is used to store the data and the most important part - the biometric scanner. The biometric scanner scans the prints of the finger of the subject and store the data in the database.

Fig. 3.1 A fingerprint scanner in use

When the subject later on place the same finger on the biometric scanner, then the biometric scanner compares the finger prints captured with the ones already present in the database. If the finger prints are matched, then the person is authorized else it alarms the personnel of a possible unauthorized person.

It has a wide variety of applications but the most common one is of recording attendance. Industries and educational institutions use these biometric scanners in order to record the attendance of the employees.

Fig. 3.2 A digital image of a fingerprint

In the educational institutions, the students will have to come and five their finger prints in the biometric scanner. Then it start identifying them the next time onwards for the attendance.

But the issue here is that the students may to stand in a queue and wait for their turn in order to give attendance which is pretty time consuming and tiring. Hence, it is not so widely used for recording the attendance of students as the more the number, the more time consuming it will be.

3.2 Iris Based Recognition System:

Iris is one of the most distinct and sensitive part of a human being in the human eye. As far as extremely sensitive and high profile materials and information is concerned, an iris scanner is used a security access tool. Highly guarded areas usually have these security access tools as a very less number of people have the permission to access them.

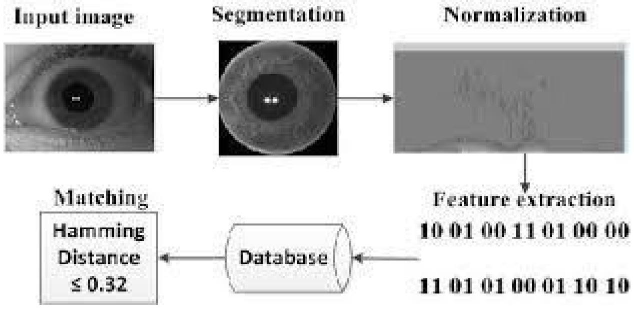


Fig. 3.3 Process of iris scanning recognition

These systems contain a scanner which scans the human eye when it is brought closer and stores the data of the iris of the human eye of the subject in the database. The error rate in this system is very low.

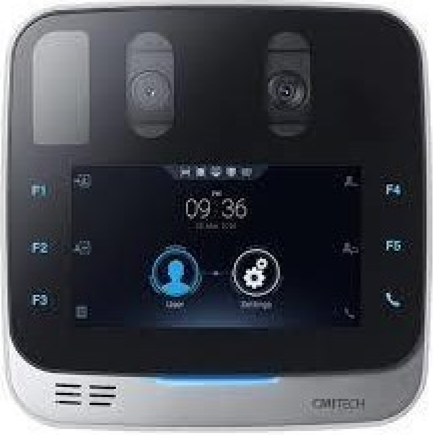


Fig. 3.4 An iris scanning machine

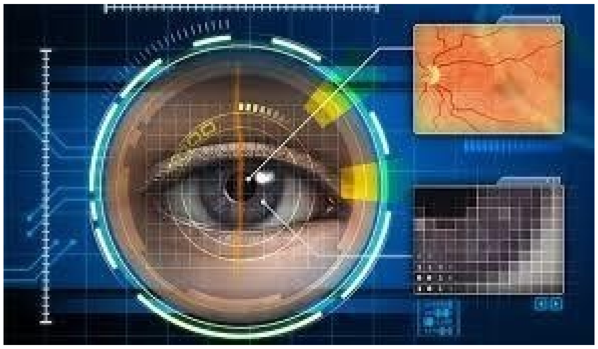


Fig. 3.5 An impression of the iris through a scanner

Since attendance is not such a sensitive information and also since every student may have to stand in a queue and give the scanner sufficient time scan the iris, it will consume a lot of time as each student make take time adjust his or position accordingly and may turn out to be not much of a good investment as it is also not so economically feasible.

3.3 PROBLEM STATEMENT

Attendance is an important aspect of daily classroom evaluation. At the beginning and ending of class, it is usually assessed by the teacher, but it may appear that a teacher may miss someone or some students answer multiple times. Face recognition-based attendance system is a problem of recognizing face for taking attendance by using face recognition technology based on high definition image capturing and other information technology.Face detection is the basic technology for human computer interaction. It can get information from faces in photos or video. The technology that recognizes the human face analyses the image of the face to extract face shape, and identify a specific target.

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. Human brain can automatically and instantly detect and recognize multiple faces. But when it comes to computer, it is very difficult to do all the challenging tasks on the level of human brain. The face recognition is an integral part of biometrics. In biometrics, basic traits of human 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 happens 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.

CHAPTER 04

PROPOSED SYSTEM

We are proposing a system which will reduce the human efforts and errors in marking attendance of every individual present in the class as well as decrease the time taken by students for answering the attendance. The proposed method also does not permit student to give proxies and reduces the bunking of classes by the students.

This is an effective way which can not only be used in schools and colleges but can also be used in various Multinational companies to mark the attendance of the employees and in various other industries as well.

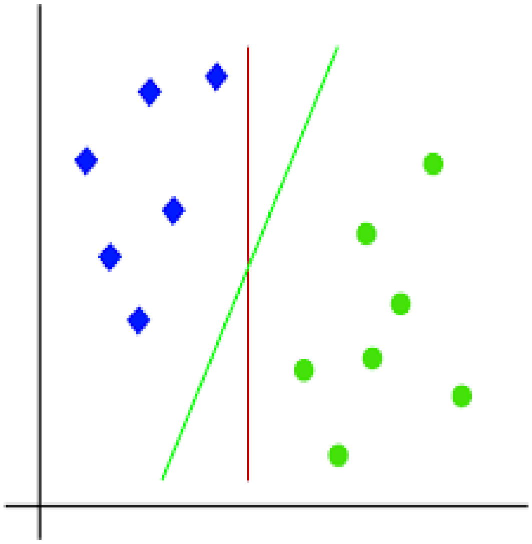
4.1 MATLAB:

4.1.1 Algorithms:

Viola–Jones:

The properties of image search require that the entire world have a set of image pixels within a rectangular area. So, they are introducing some of the Haar functionality, which has been used previously in the research of digital objects. However, since the characteristics of Viola and Jones are based on more than four dimensions, they are undeniably contradictory. The number to the right shows four different designs used. The advantage of one given to them is to group pixels into clear rectangles and subtract those pixels into four squares. Definitions of this type are usually lower compared to other materials such as steerable filters. Although they do have side effects and horizontal, the answer is very clear.

SUPPORT VECTOR MACHINE:



Data segmentation by determining the decision size (hyperplane) separates all the points in a class other classes. The best hyperplane for SVM is the one that has the most learning between these two classes, when data has rows and divisions. If the data is not in a split line, a other labor of torture is wrongly employed hyperplane. SVMs sometimes use a kernel to modify ambiguous data is then distributed in a high-resolution manner high resolution can be achieved.

Fig. 4.1 Support Vector Machine

4.1.2 Working:

There are majorly 3 sections in this project, that are:

1) Face recognition
2) Face verification
3) Attendance monitoring

Face recognition:

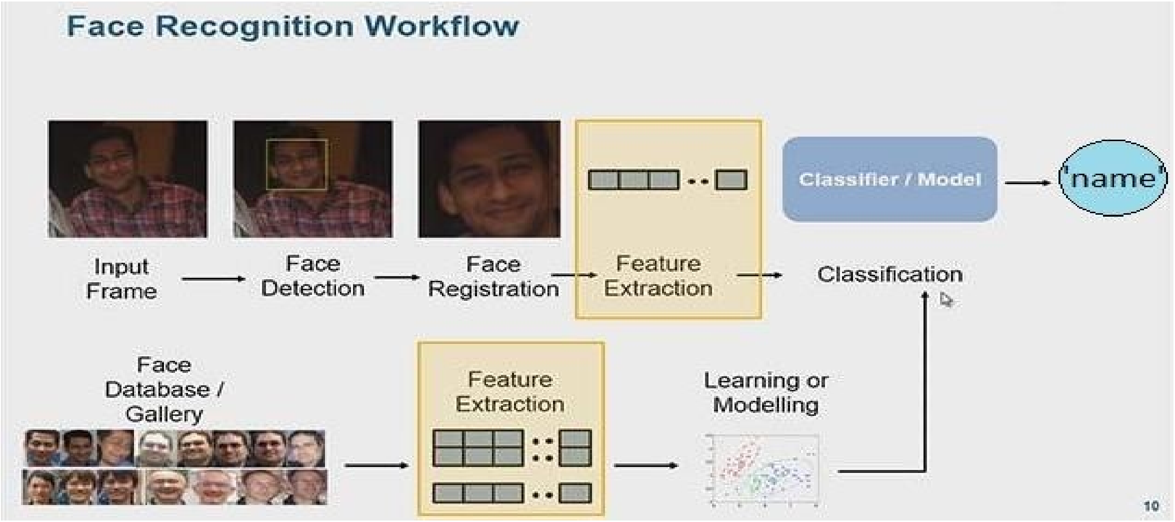


Fig. 4.2 Process of facial recognition using MATLAB

The first step in the face recognition is input frame which is took by the camera. After the input frame is captured that image will go for face detection, for face detection we used the algorithm "**Viola Jones**"**.** If the face is present in the input frame then the frame is cropped where the face is present. Then the next step is to Feature Extraction, Feature extraction a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector. This approach is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as image matching and retrieval.

For our projects we used HOG feature extraction. The histogram of arranged angles (HOG) is a component descriptor utilized in computer vision and picture handling with the end goal of item location. The method includes events of inclination direction in limited segments of a picture. This strategy is like that of edge direction histograms, scale-invariant component change descriptors, and shape settings, yet varies in that it is figured on a thick lattice of consistently divided cells and utilizations covering nearby complexity standardization for improved precision.

The next step is to classify according to predefined model. The model is created from the face database where database should contain the faces of students in the a particular folder where the images in the database go for the feature extraction and then the feature extraction will go for the learning model hear we used "**support vector machine**" as a learning model, "Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used utilized for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate.

The output for the classification of the input frame will give the name of the person which will matched from the learning model.

Face verification

Face verification process is almost same has face recognition, here we will take the HOG futures image which is captured and this and HOG futures which is recognized and we will take the difference between those futures. This future will pass to the predefined model. If those belong to same person it will give 1 If the images are belong to different person then the classifier will give output as 0.

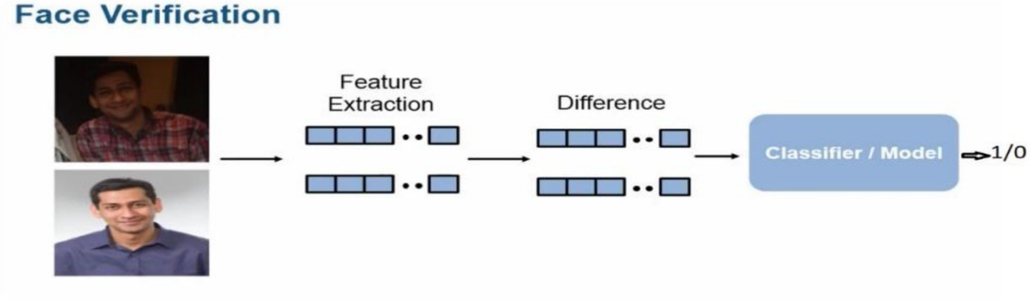


Fig. 4.3 Process of face verification using MATLAB

The model for the face verification, we will take the two images of the same person and extract HOG futures and take difference between those futures and the we will give the value as 1, and then we will take the two different persons and difference between those HOG futures and give the value as 0. This will be pass to the SVM Learning model.

Attendance monitoring

Attendance will be monitored on the hourly basic. The camera will be turned on for the give time. If there are any holidays and Sundays the camera will be turned off. When camera is done the it will take the images continually and go for face recognition and face verification. If the person is matched with the database then the attendance will be marked as 1 for that give person. If the person is absent then it will be marked as 0 this format will be saved to excel sheet according to class hour. Each excel sheet for each hour it will be saved in the folder with the time. Each time folder is save in a folder with the text of date of particular day. This all folder will saved in the attendance folder. Excel sheet will be created after each class.

4.1.3Face recognition work flow

- Open image database collection=imageSet( )
- Extract futures in images futures=extractHOGFeatures( )
- Fit the model using machine learning class=fitcecoc( )
- Intialinsing computer vision FaceDetect=vision.CascadeObjectDetector
- Detect face in the image b=step(FaceDetect,img)
- Detect the face from the model name=predict(class,features1)
- Output: attendance is marked in Excel sheet

4.2 Python

4.2.1 Algorithm:

Haar Cascade:

Haar Cascade is basically a classifier which is used to detect the object for which it has been trained for from the source.

The haar cascade is trained by super imposing positive images over a set of negative images. The training is generally done on a server and on various stages.

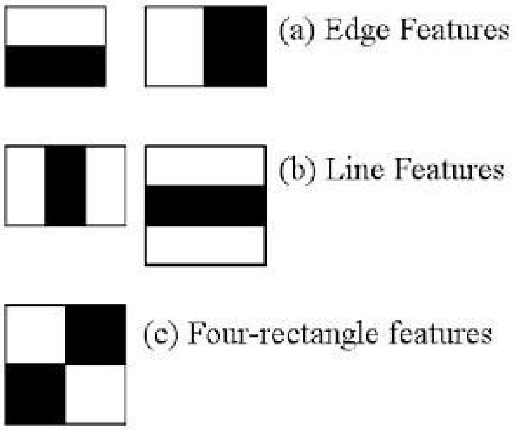
Here we will be using haarcascade_frontal_default.xml file to help us with faster facial recognition.

Haar cascade frontal face default is a machine learning based approach where a cascade function is trained from a lot of positive and negative images.

This algorithm has four stages:

- Haar feature selection
- Creating integral images
- Adaboost training
- Cascading classifiers

At the initial stage, the algorithm has to have a lot of positive and negative images without faces to train the classifier. We then start to extract features from these positive and negative images.



The very first step here will be to collect haar features. Haar features always considers adjacent rectangular regions in a detection window at very specific location. This will basically sum up the intensities (pixel count) of pixels in every region and will later calculate the difference between these sums.

Fig. 5.4 Haar Cascade Algorithm Features Extraction types

To make the process faster here we will use integral images.

Among everything calculated we will find that most of it is irrelevant. For example, consider the images shown below. The row on top shows two features that are good. The first feature selected seems to focus on the property that the region of the eyes compared to that of the cheeks and nose is much darker. The second feature relies on the property that the eyes are much darker compared to the bridge of the nose.

The same applied on the cheeks and any other place becomes irrelevant.

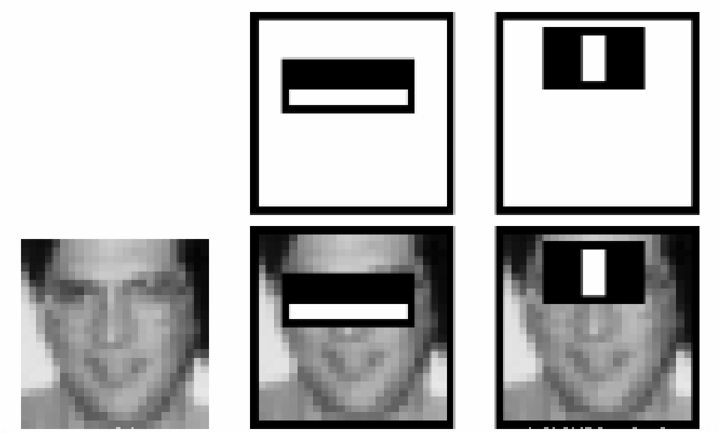


Fig. 4.5 Usage of the feature on the image

Now the question arises as to how does the selection of the best features out of 160000+ features happen? This task is accomplished by using a concept called Adaboost.

Adaboost will select the best features and will also train the classifiers that happen to be using them. Thus this specific algorithm will therefore construct a “strong” classifier which is a linear combination of the weighted simple “weak” classifiers. The process is as follows:

During the phase of detection, over the input image a window of the target size is moved and the haar features are calculated for every subsection of the image. This difference is later compared to a learned threshold that separates objects and non-objects. As each Haar feature is only a "weak classifier" (its detection quality is slightly better than random guessing) a large number of Haar features are necessary to describe an object with sufficient accuracy and are therefore organized into cascade classifiers to form a strong classifier.

Cascade Classifier:

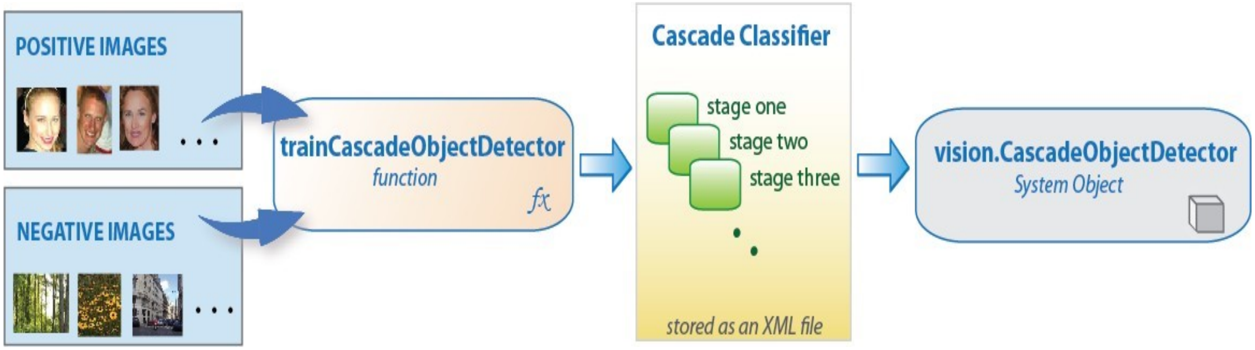
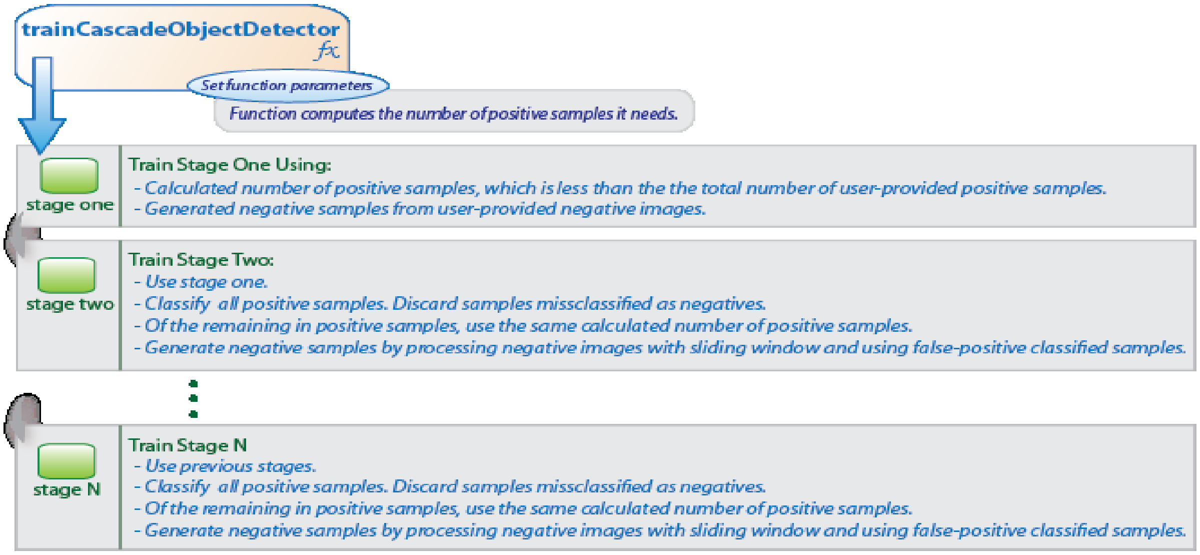


Fig. 4.6 Cascade Classifier

There are a collection of stages in the cascade classifiers in which every stage is an ensemble of the weak learner. The weak learner are simple classifiers called decision stumps. Every stage is trained using a technique called boosting.

This will provide the ability in training a highly accurate classifier by taking a weighted average of the decision made by the weal learner.



Function computes the number of positive samples it needs.

Train Stage One Using:
- Calculated number of positive samples, which is less than the the total number of user-provided positive samples.
- Generated negative samples from user-provided negative images.

Train Stage Two:
- Use stage one.
- Classify all positive samples. Discard samples missclassified as negatives.
- Of the remaining in positive samples, use the same calculated number of positive samples.
- Generate negative samples by processing negative images with sliding window and using false-positive classified samples.

Train Stage N
- Use previous stages.
- Classify all positive samples. Discard samples missclassified as negatives.
- Of the remaining in positive samples, use the same calculated number of positive samples.
- Generate negative samples by processing negative images with sliding window and using false-positive classified samples.

Fig. 4.7 Process of Cascade Classifier in Python

Face Recognizer:

OpenCV has three built in face recognizers and because of the openCV's clean coding, we can use any of them just by changing a single line of code.

Below are the names of those face recognizers:

- EigenFaces face recognizer:

  cv2.face.create Eigen FaceRecognizer()

- FisherFaces face Recognizer:

  cv2.face.create Fisher FaceRecognizer()

- LBPH face Recognizer:

  cv2.face.create LBPH FaceRecognizer()

In this code we will be using LBPH (Local Binary Pattern Histogram) face recognizer.

LBPH algorithm:

LBPH stands for Local Binary Pattern Histogram, a basic algorithm that's used to detect faces from the front side. It is used for object as well as face detection. The LBP operator helps to get local features by Local Binary Pattern act . The local special arrangement of the face is shortened by these LBP acts. The LBP operator divides the face in the image into pixels. Every pixel is associated with 8 neighbor pixels that surrounds it . Each pixel value is then compared with the surrounding neighbor pixel values. The equation is for this is:

$$\text{LBP}(xc ,yc) = \sum 7 \; n=0 \; s(in - ic)2n$$

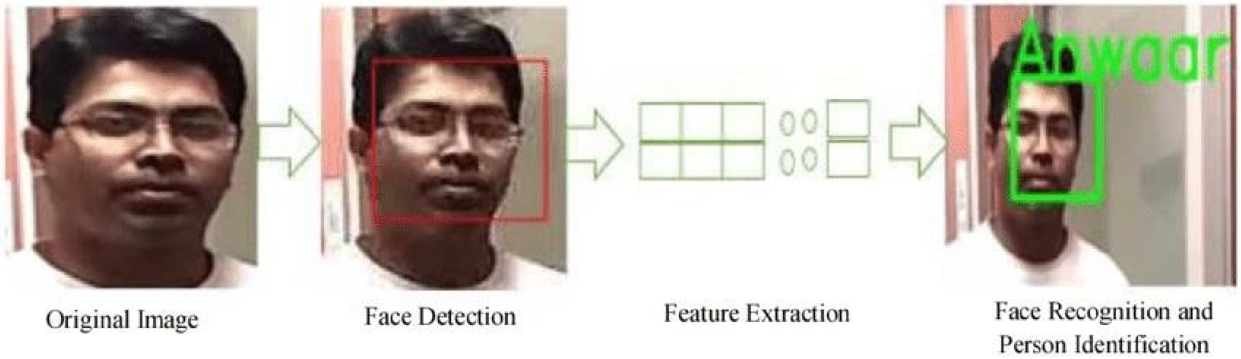


Fig. 4.8 LBPH Face Detection

4.2.2 The face recognition algorithm workflow:

Input: A person's face.

Output: Attendance is marked for recognized faces.

Initialization

1. Open the camera cv2.VideoCapture(0)
2. Import the face classifier LBPHFaceRecognizer_create()
3. Read the trained data from the file recognizer.read('trainer.yml')

GPU Part

1. Image is captured frame by frame img = cap.read()

2. Face in the frame is converted to greyscale gray = (img, cv2.COLOR_BGR2GRAY)
3. All faces are detected in the frame faces = face_cas.detectMultiScale()
4. for coordinates of face:

- values inside frame are read roi_gray = gray[y:y + h, x:x + w]
- values are compared with stored values confidence = recognizer.predict(roi_gray)

5. if confidence<85:

   face matching with the values is recognized attendance is marked

   dataEntry(name,class_count)

6. else:

   face is not recognized id = 'Unknown.

4.2.3 Implementation

1. **Pre-Processing Images:**

   The system captures around 50 images of every individuals face. The images are converted into grey scale as LBPH operates using images in greyscale and the images are stored in a folder. The stored images will be saved with a name and ID unique to that person.

2. **Face Detection:**

   When a person appears in front of the camera, the camera detects that a face by use of **"Haar cascade"** algorithm if face is present a frame appears around the face. The entire frame is converted to greyscale as LBPH works only on greyscale images. A scale factor is used to compensate for multiple faces present in front of the camera.

3. **Feature Extraction:**

   The LBPH algorithm makes use of binary values and stores the data in a file. The binary values are different for each face. The Region of Interest (ROI) are parts of the face from where features are extracted. Information about the gradients in the face is captured. The image of a person's face is divided cells comprising of 8 pixels. Each pixel present has a gradient and compares itself with its neighbor pixels.

4. **Face Recognition:**

   In the comparison module, face recognition process is carried out. When a face is detected by the camera it checks the corresponding values of the current visible face with values stored in the file. If the values are a match, then the face is recognized and the name associated with that face is displayed.

5. **Attendance database(excel)**

Database holds the name and ID of all people whose attendance will have to be marked. As and when a face is detected and matched with the existing records, the attendance is automatically updated in the database. Every time a face is recognized, the number of times that person is present increases by a count of 1. There is also a column which shows when the attendance was last updated.

BLOCK DIAGRAM

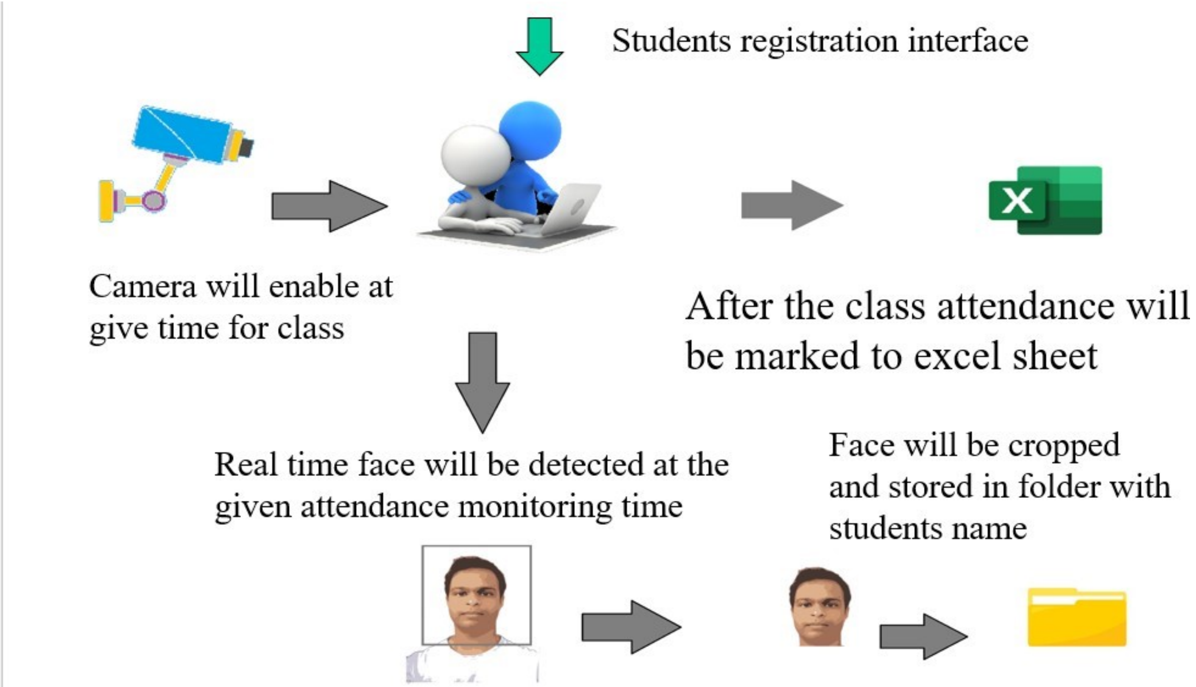

Students registration interface
Camera will enable at give time for class
After the class attendance will be marked to excel sheet
Real time face will be detected at the given attendance monitoring time
Face will be cropped and stored in folder with students name

Fig. 4.9 Block Diagram

CHAPTER 05

HARDWARE AND SOFTWARE SPECIFICATIONS

For the completion of the proposed project we need to meet certain specification of both the hardware and software. The specifications are as follows:

Hardware specifications:

Camera:

Fig. 5.1 Camera

A camera is an optical instrument used to record pictures. At their generally fundamental, cameras are fixed boxes with a little opening that let light in to catch a picture on a light-touchy surface (typically photographic film or a computerized sensor). Cameras have different systems to control how the light falls onto the light-touchy surface. Focal points center the light entering the camera, the size of the opening can be broadened or limited to give pretty much light access to the camera, and a screen system decides the measure of time the photograph delicate surface is presented to the light.

SOFTWARE SPECIFICATION:

MATLAB

MATLAB APP DESIGNER

Matlab is a programming environment. Matlab is used in many technical areas for data analysis, problem solving and experimentation and algorithm development. Discipline-specific software, written in Matlab, is divided into libraries of functions called toolboxes, which are widely used. Matlab has found widespread use in technical education as a basis for computational laboratory work.

Fig. 5.2 MATLAB app designer

The main reasons for Matlab's success are its intuitive, concise syntax, the use of complex metrics as the default numerical data object, the power of built-in operators, easy-to-use graphics, and general agreement in its technical computing community. Simple and user friendly editing environment, allowing easy extension of the language. This can add to the reliability of numerical methods based on operators.

Matlab appdesigner

App Designer is an interactive development environment for designing an application layout and programming its behavior. It provides a fully integrated version of the MATLAB editor and a large set of interactive UI components. It also offers Grid Layer Manager to organize your user interface and automatic reflow options to detect and respond to changes in your application screen size. The App Designer allows you to create a professional software developer without the need for professional applications. Use a graphical user interface (GUI) and an integrated editor to quickly program its behavior.

Share your apps using Matlab Drive, or create standalone desktop or web applications with the Matlab Compiler Sim and Simulink Compiler.

Python:

Libraries Required:

- OpenCV-Python:
- Pandas
- Numpy
- Openpyxl
- Tkinter

Fig. 5.3 PYTHON 3.8.0

OpenCV python:

OpenCV is an image and video processing library with bindings in C++, C, Java and Python. OpenCV is used for all sorts of image and video analysis like facial editing, advanced robotic vision, optical character recognition and a whole lot more.

Fig. 5.4 OpenCV PYTHON

Fig.5.5 Pandas

Pandas creates rows and columns called data frames. This will look very similar to the tables in any statistical software.

Here it converts xlsx which is the file format for Microsoft Excel to CSV (comma separated values) which is a very simple file format used to store tabular data such as spreadsheets. Pandas can very easily store the values which here is attendance in CSV format and then convert it back to xlsx format which is Microsoft excel.

Numpy:

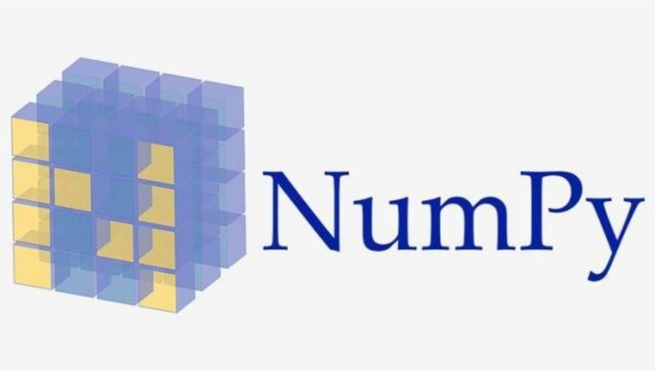
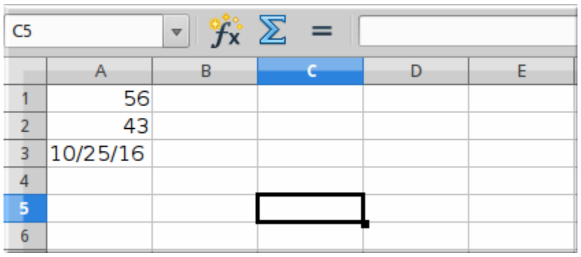


Fig. 5.6 Numpy

Numpy is the python library which is a python package of general purpose array processing. Numpy provides multi-dimensional array object which are high performance and tools for working with the specified arrays.

Openpyxl:



	A	B	C	D	E
1	56				
2	43				
3	10/25/16				
4					
5					
6					

Fig. 5.7 Openpyxl

Openpyxl is a python package which helps to work with excel.

Tkinter:

Fig. 5.8 Tkinter

This is a python package which works similar to HTML. Instead of importing the code for HTML into python, this package helps us to create the similar page with much more simplified language. Things which can be done on HTML can be easily done in python using this package.

CHAPTER 06

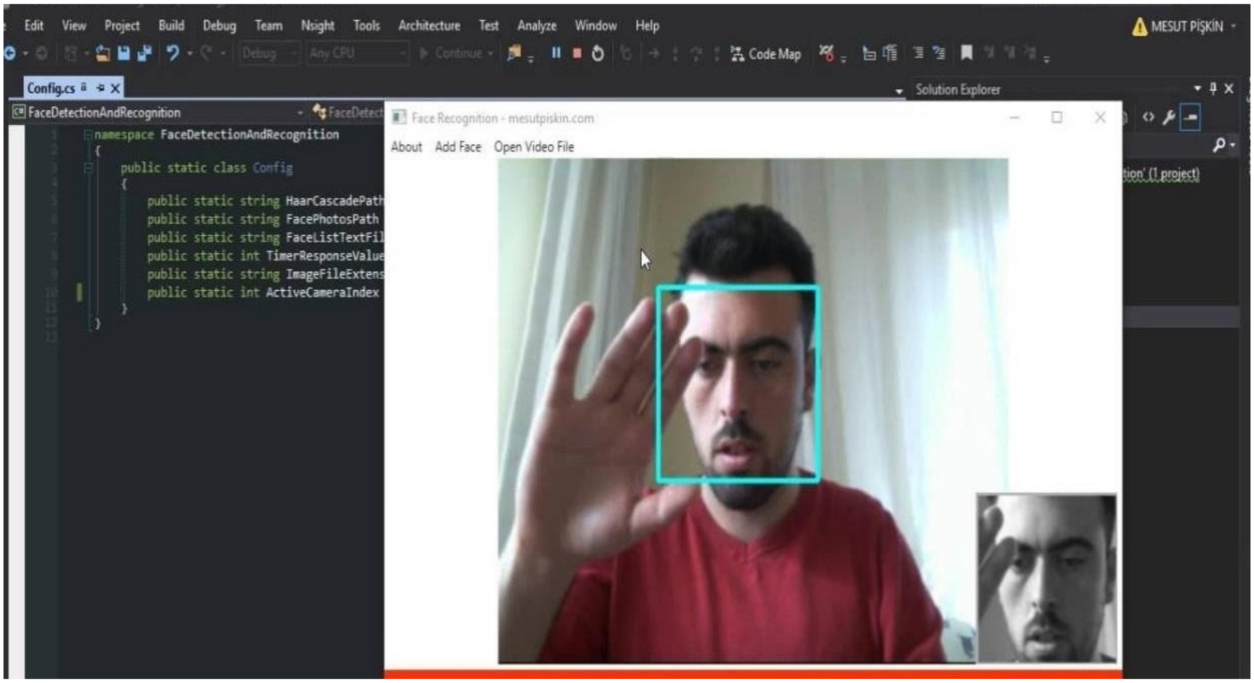
RESULT AND DISCUSSION

The result which we got after the project is that the image captured by the fixed camera is detected and recognised. Further the recognized image of the student is provided with attendance, if it matches with the database of the entries which are already provided else the system marks absent.

The Camera is constantly fixed at a specific distance inside a classroom to seize frontal photographs of the students inside the classroom. After the completion of detection and processing the face, it is compared to the faces present inside the student's database to update the attendance of the students.

Monitoring is one of the major responsibilities of a teacher. Teachers have a lot of things to do in their daily meetings and taking hands-on work takes up a lot of quality time that they have to offer in teaching. The new system will provide an easy way for teachers to monitor students. So we created an automated monitoring system for students using facial recognition.

The manually attendance covers the problem of using a manual system such as unsecured record of attendance, multiple paper operations, and the accuracy of data input. This proposed system that utilizes facial recognition in attendance will assist institutions with their development and improvement through the use of new technologies and an improved system of daily attendance procedures.



Edit View Project Build Debug Team Nsight Tools Architecture Test Analyze Window Help
MESUT PIŞKIN
Debug
Any CPU
Continue
Code Map
Config.cs
Solution Explorer
FaceDetectionAndRecognition
Face Recognition - mesutpiskin.com
About
Add Face
Open Video File
tion' (1 project)

```
namespace FaceDetectionAndRecognition
{
    public static class Config
    {
        public static string HaarCascadePath
        public static string FacePhotosPath
        public static string FaceListTextFil
        public static int TimerResponseValue
        public static string ImageFileExtens
        public static int ActiveCameraIndex
    }
}
```

Figure 6.1- Capturing of the face using camera

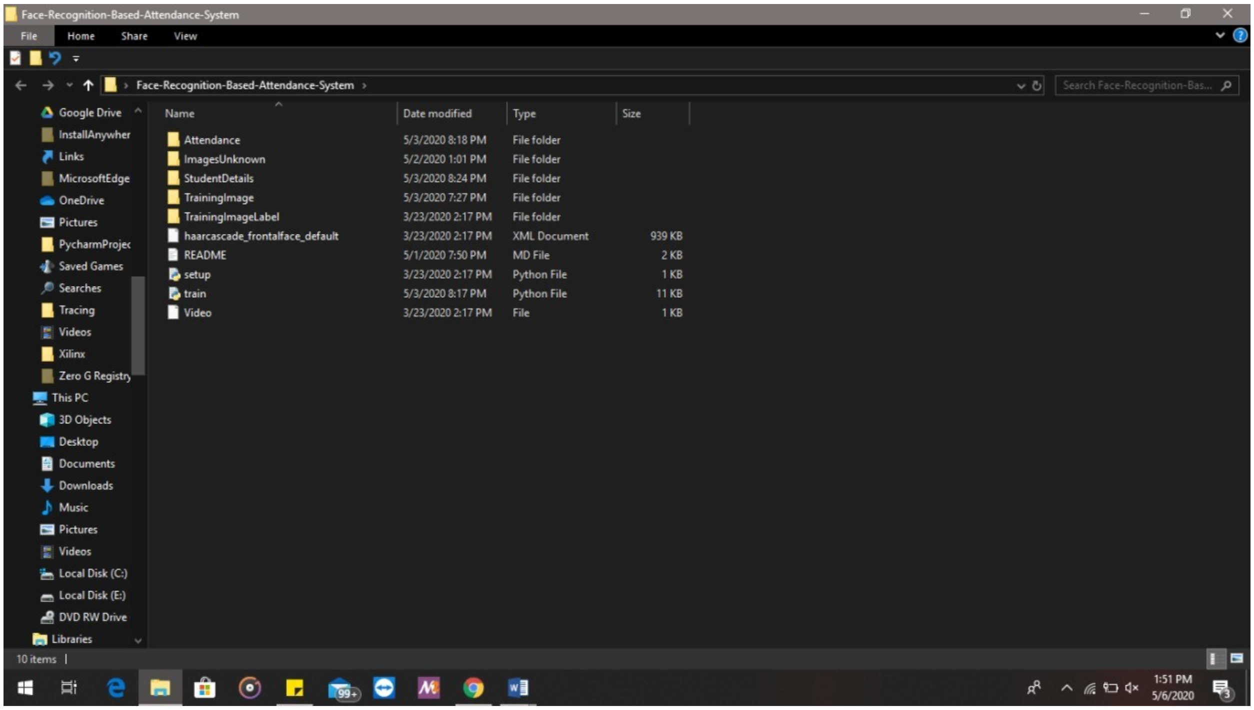
The above image shows how the frontal image of the student is being taken by the camera. And later this image will be processed and analysed. When this image is matched with any of the images present in the database the attendance will be marked otherwise it will be marked absent.

In the database for a single student several pictures of his face are already taken as a sample which also increases the accuracy of detection of the faces and monitoring attendance becomes accurate and easy for the teachers.

Due to so many sample pictures of a single student it is not possible for other student to mark proxy and the system also does not confuse a face over other student as a result the accuracy enhances as compared to manually attendance system.

Python

We have used a source code editor developed by Microsoft knows as VS Code (Visual Code Studio). This works just like the python IDE but has added advantages to it.



Name	Date modified	Type	Size
Attendance	5/3/2020 8:18 PM	File folder	
ImagesUnknown	5/2/2020 1:01 PM	File folder	
StudentDetails	5/3/2020 8:24 PM	File folder	
TrainingImage	5/3/2020 7:27 PM	File folder	
TrainingImageLabel	3/23/2020 2:17 PM	File folder	
haarcascade_frontalface_default	3/23/2020 2:17 PM	XML Document	939 KB
README	5/1/2020 7:50 PM	MD File	2 KB
setup	3/23/2020 2:17 PM	Python File	1 KB
train	5/3/2020 8:17 PM	Python File	11 KB
Video	3/23/2020 2:17 PM	File	1 KB

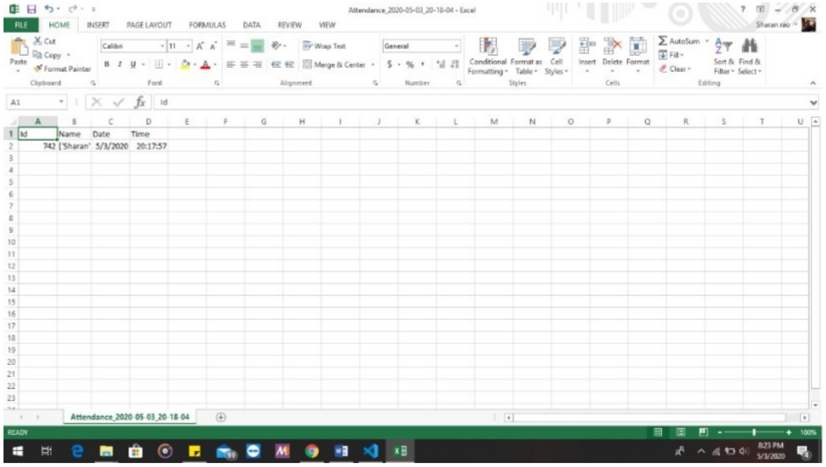
Fig. 6.2 The entire set up of the software in a folder

We have included a couple of folders in the directory to better functioning of the code. They are:

- Attendance
- Images unknown
- Student details
- Training images
- Training images label
- Haar cascase frontal default.xml
- Setup.py
- Train.py

Now each folder is designed in a specific way to carry out specific functions.

Attendance:



Attendance_2020-05-03_20-18-04 - Excel

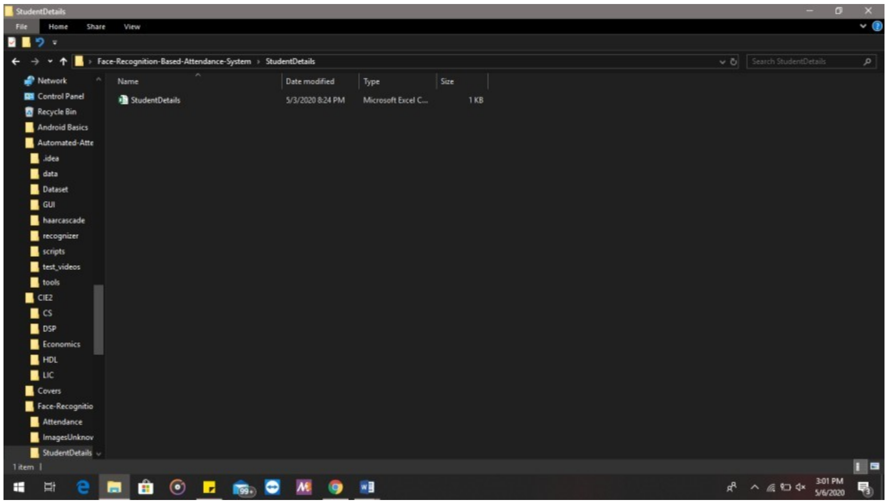
Id	Name	Date	Time
742	['Sharan'	5/3/2020	20:17:57

Fig. 6.3 Attendance recorded in a spreadsheet

This folder contains an excel sheet in which attendance of each student will be recorded.

Images unknown: This folder contains images captured by the camera which it doesn't recognize or which were not trained or objects which can be used by the developer for any further use.

Student details: I have made sure that the attendance report of every individual student is available on excel sheet separately for further use of either the parents, teachers or the students.



Face-Recognition-Based-Attendance-System › StudentDetails

Name	Date modified	Type	Size
StudentDetails	5/3/2020 8:24 PM	Microsoft Excel C...	1 KB

Fig. 6.4Student Details in a separate Excel Sheet

Training Images:

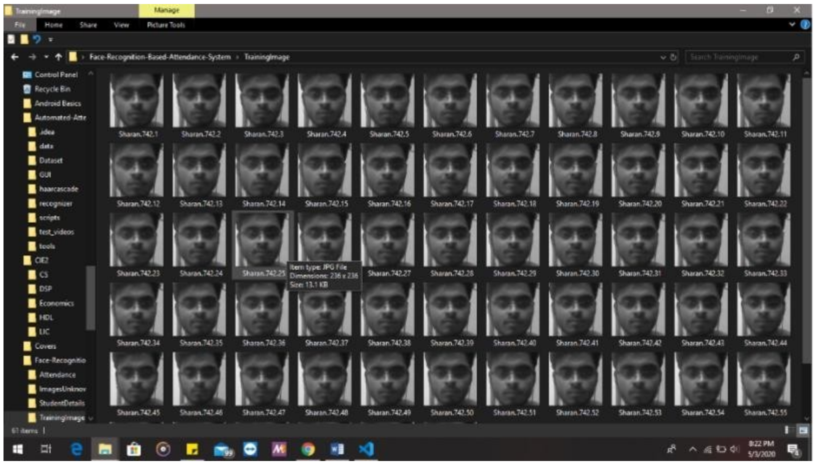


Fig. 6.5The images taken by the camera

It is mandatory for every student to store his/her image in the database for the purpose of recording attendance. All the images captured will be trained by the compiler and stored in this folder.

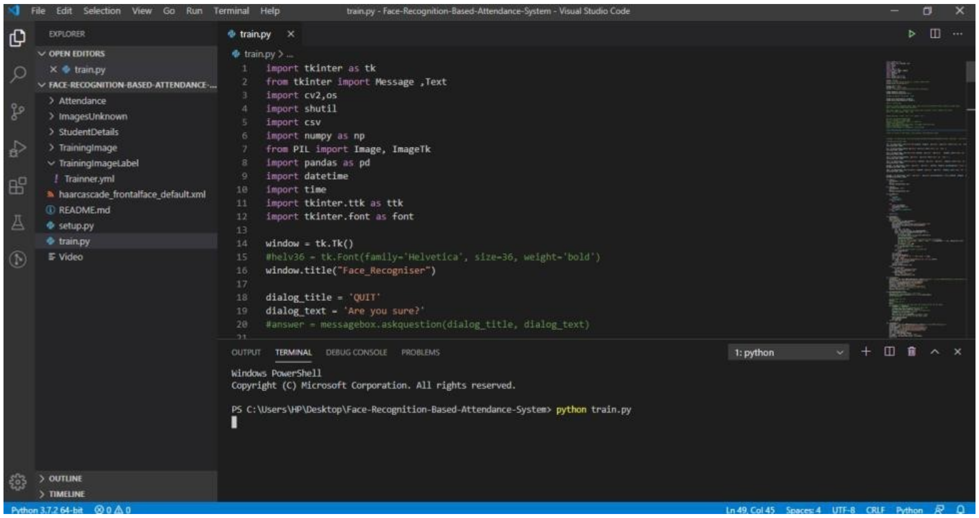
Training Images Label: This is a basic algorithm that helps us in training the images.

Haar cascade algorithm: This algorithm helps in distinguishing between faces trained and helps in proper maintenance of attendance.

Setup.py: This is the python code which helps in basic assembling of the code and helping bring together all the different parts of the directory alive and functioning smoothly.

Train.py: This is the brains behind the functioning of the project. All the major logic is applied into this so that we get the desired output. Here we can change and reprogram our code based on our specification.

Code functioning:



```python
import tkinter as tk
from tkinter import Message ,Text
import cv2,os
import shutil
import csv
import numpy as np
from PIL import Image, ImageTk
import pandas as pd
import datetime
import time
import tkinter.ttk as ttk
import tkinter.font as font

window = tk.Tk()
#helv36 = tk.Font(family='Helvetica', size=36, weight='bold')
window.title("Face_Recogniser")

dialog_title = 'QUIT'
dialog_text = 'Are you sure?'
#answer = messagebox.askquestion(dialog_title, dialog_text)
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\HP\Desktop\Face-Recognition-Based-Attendance-System> python train.py
```

Fig. 6.6 Beginning of the functioning of the code

To start, we will have to type in a command in the terminal window which will start the execution. This will open up a separate window where one can enter his details and train his image after capturing it with the help of the camera.

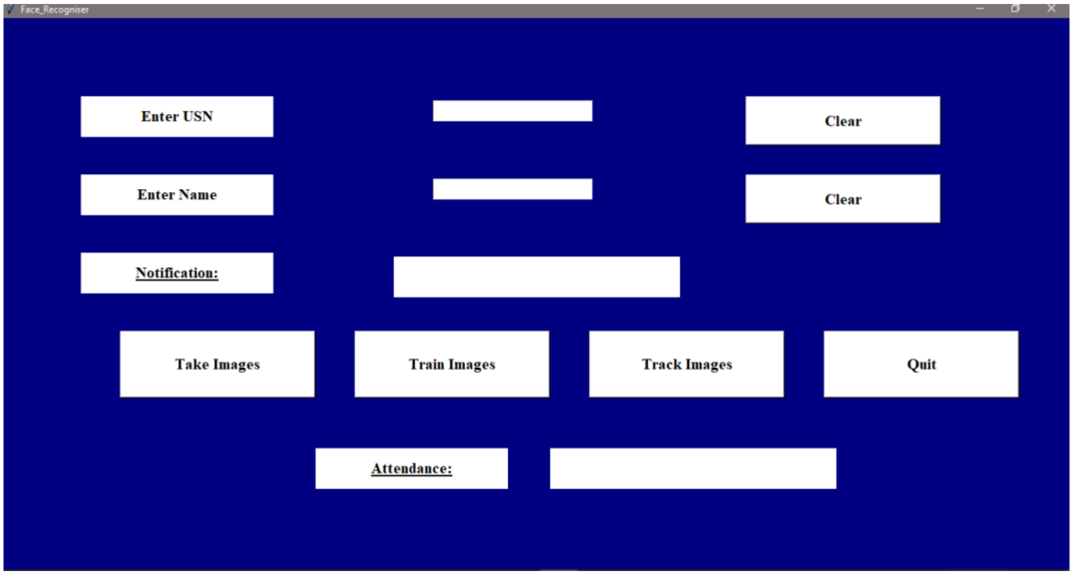


Fig. 6.7 Template of Student Details Entry

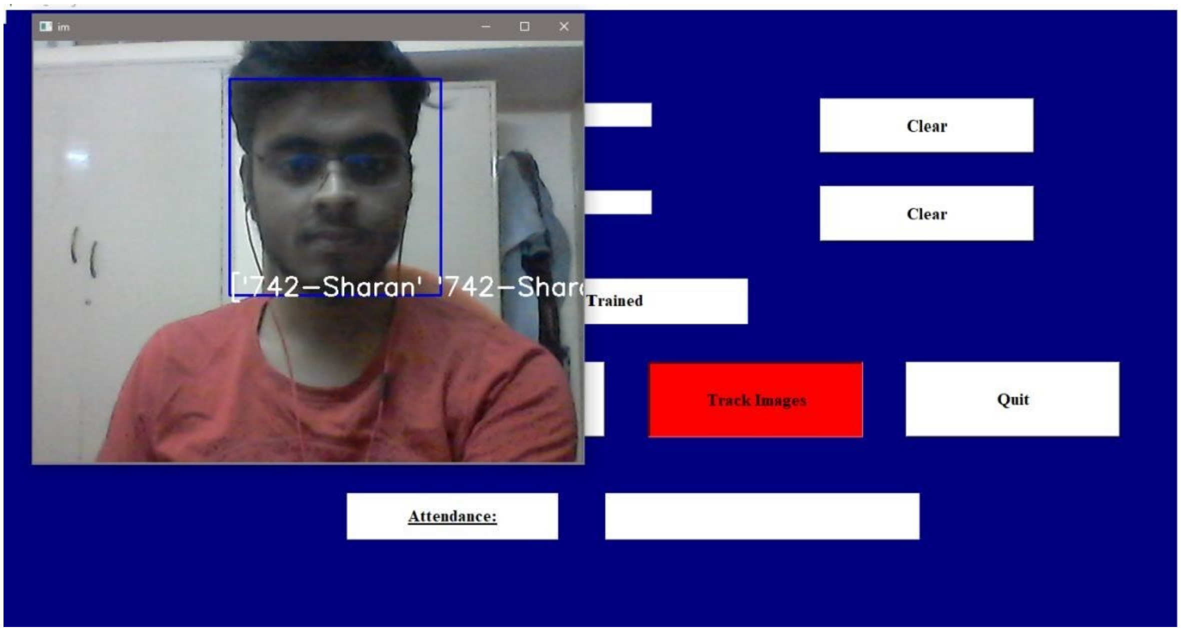


Fig. 6.8 Image tracking and recording of attendance of the student

The take image option will capture a number of images required for training the images for better facial recognition. Once this is done the student needs to only sit in front of the camera and track his picture and the algorithm will detect his face and make the entries in the excel sheet. For verification purpose it will show the student his/her name so that wrong entries would not be marked on the excel sheet.

Matlab:

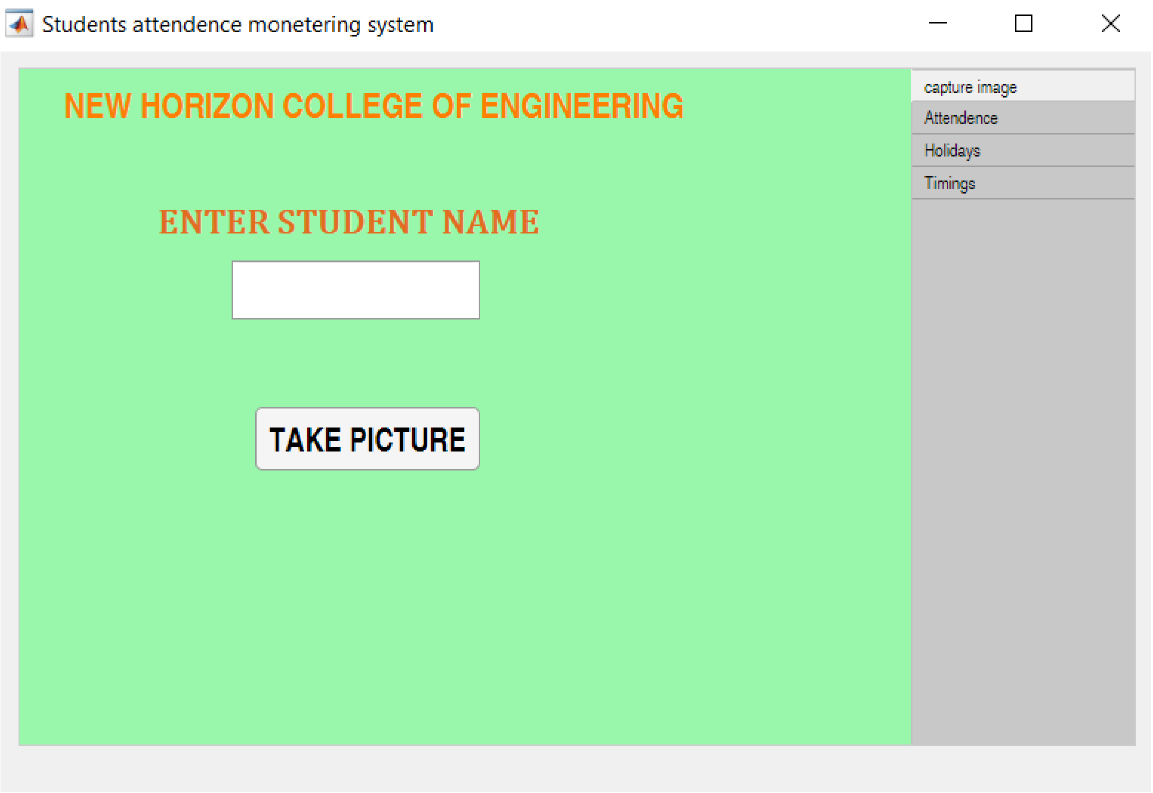

Students attendence monetering system
NEW HORIZON COLLEGE OF ENGINEERING
ENTER STUDENT NAME
TAKE PICTURE
capture image
Attendence
Holidays
Timings

Fig. 6.9Student Details Entry

The overview of Graphical user interface in MATLAB for Students Attendance Monitoring System. The image (above) shows the student's registration interface where the student can enter the name of the student and then if we click on the 'Take Picture' , the student's photos of the given number will be captured and it will be moved to the particular folder.

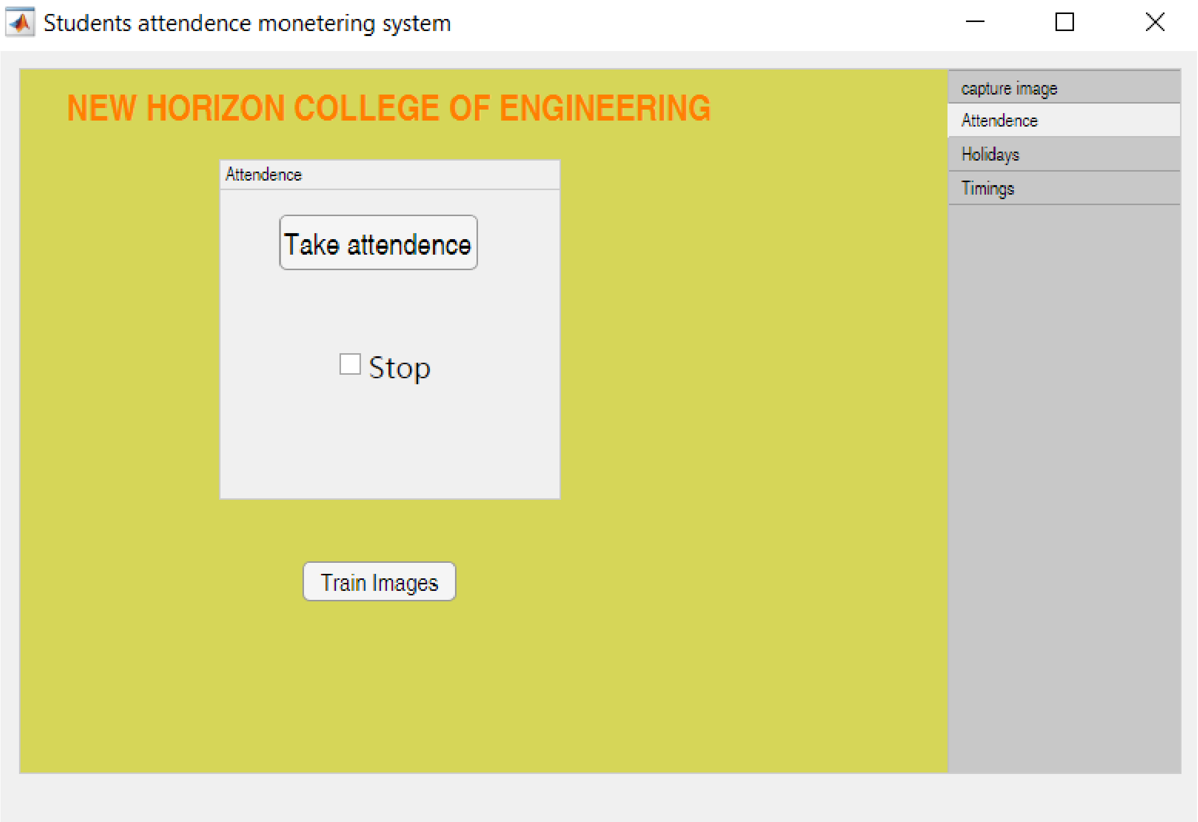
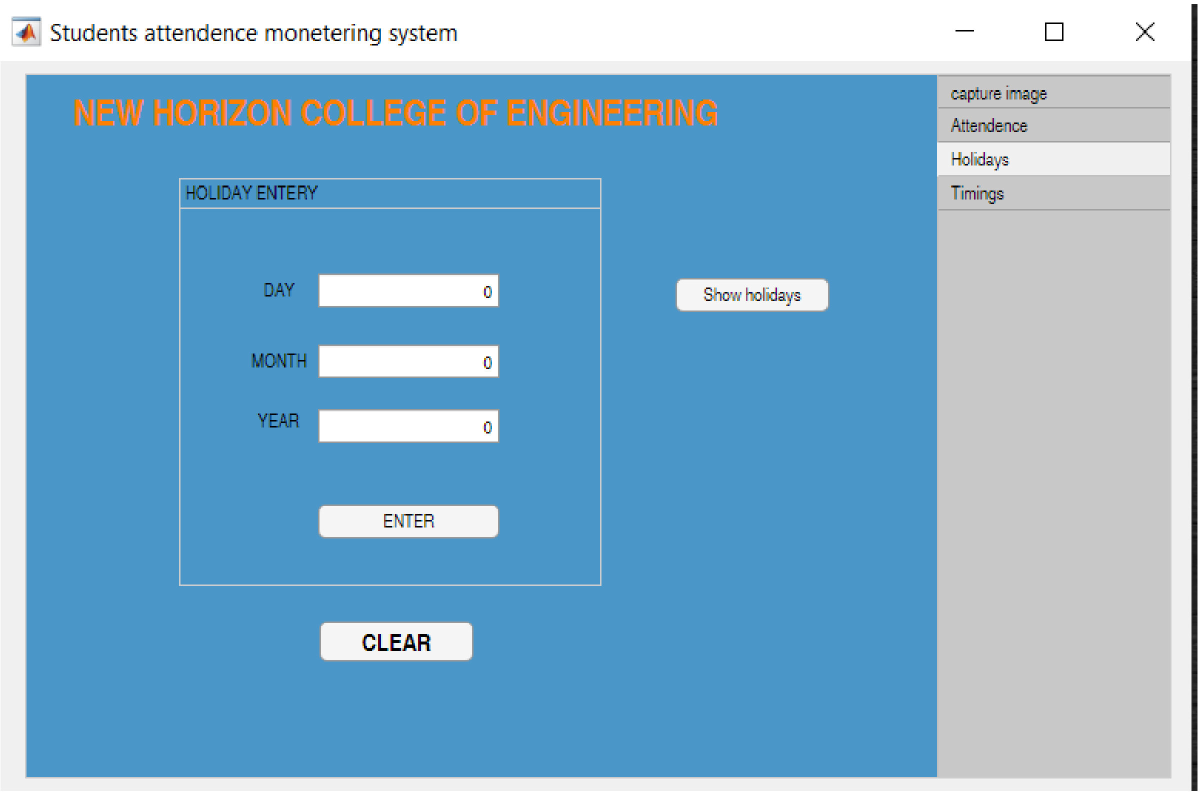


Fig. 6.10 Template of taking the image of the student

This is the interface to start the attendance. Before taking the attendance, we should train the captured image. Then we can start taking the attendance. If we need to stop the attendance we can click on stop.



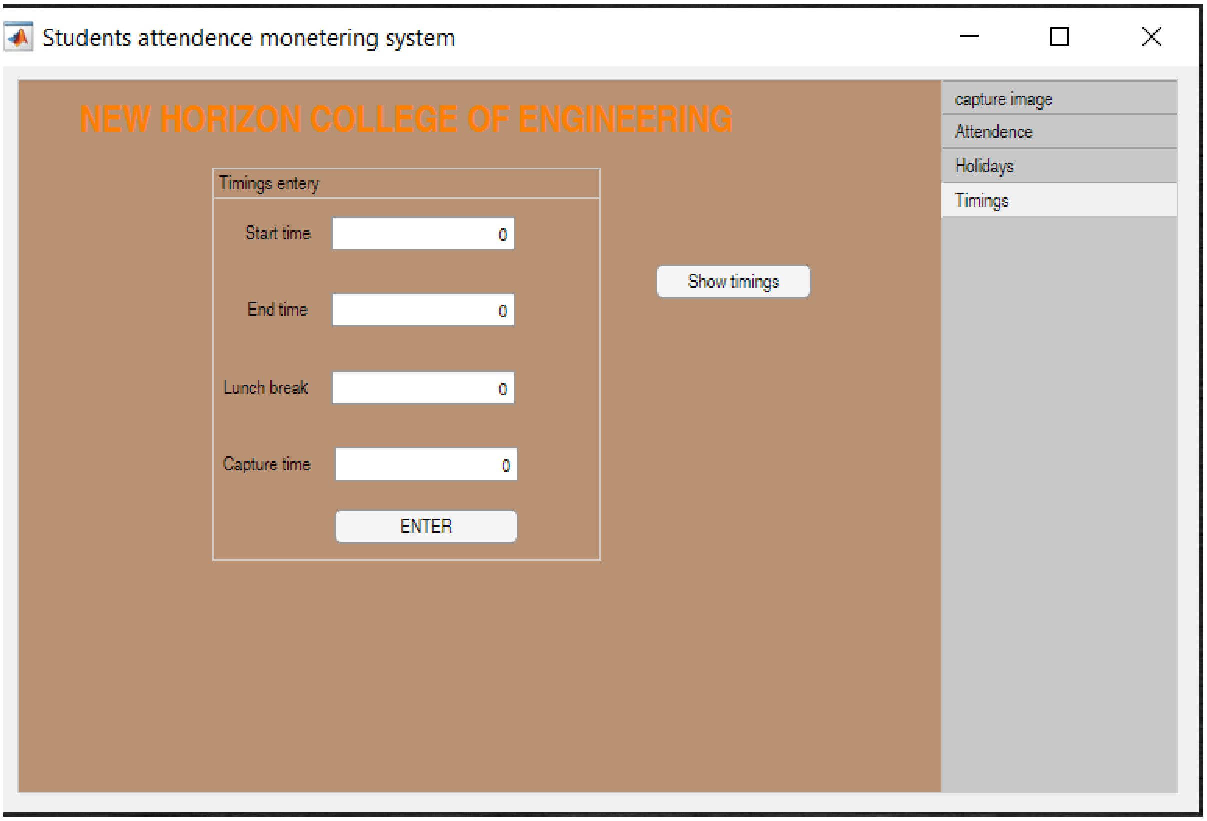
DAY 0

MONTH 0

YEAR 0

Fig. 6.11 Template to enter the holidays

This is the interface for entering the holidays, on the entered date the attendance will not be taken. We can enter n number of holidays. After entering the holidays, we need to press on Enter. If we need to clear holidays we can use clear button. The Show Holidays button is used to show the entered holidays in the MATLAB command window.



Students attendence monetering system
NEW HORIZON COLLEGE OF ENGINEERING
capture image
Attendence
Holidays
Timings
Timings entery
Show timings
ENTER

Start time 0

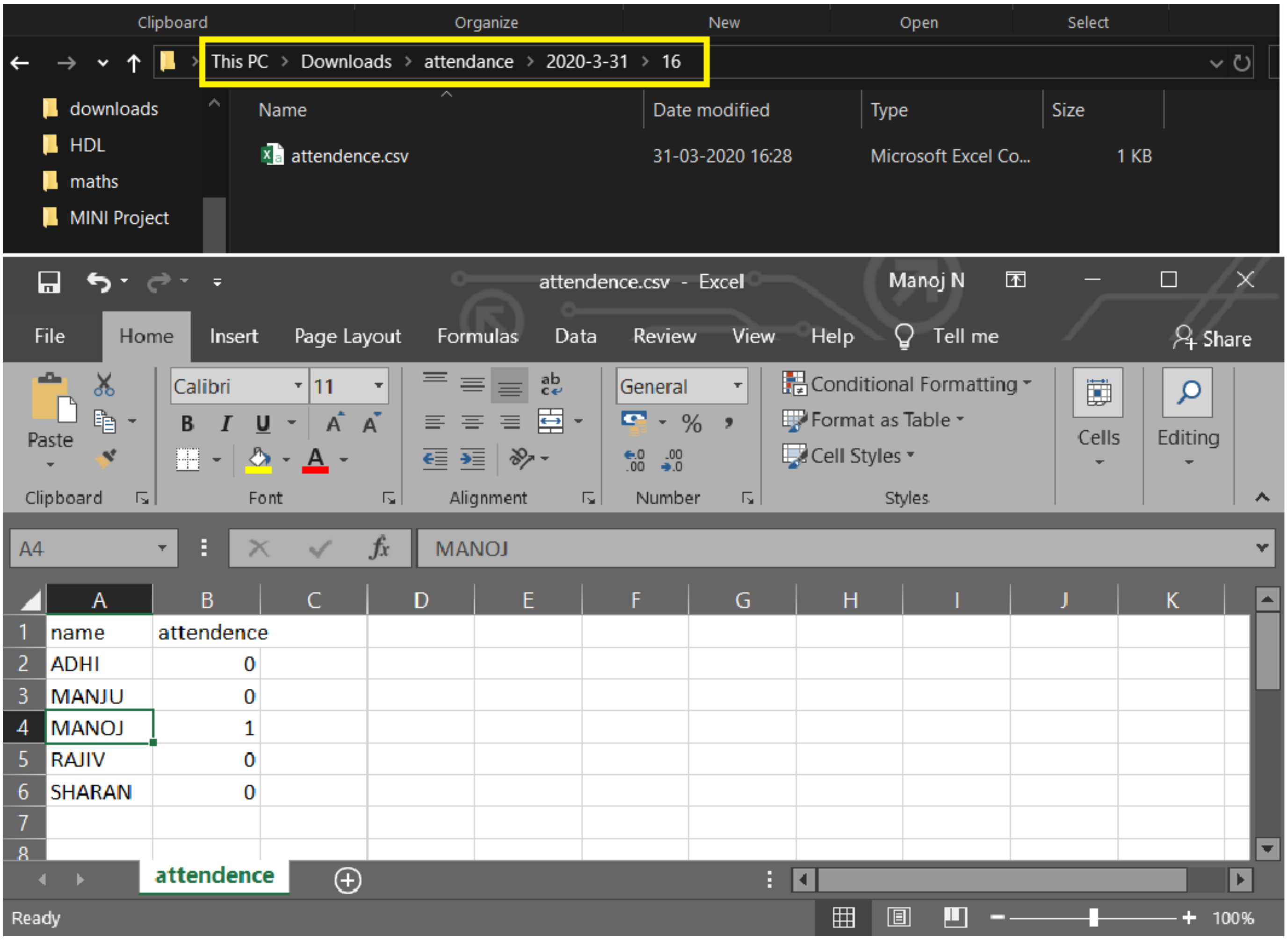
End time 0

Lunch break 0

Capture time 0

Fig. 6.12 Template to enter the timings of the classes

The above image shows entry of timings, where we can enter the start time, end time, lunch time, and how long the attendance is required to be taken for the given class. For the given capture time, the camera will be enabled and the images will be considered for attendance.



This PC > Downloads > attendance > 2020-3-31 > 16

Name	Date modified	Type	Size
attendence.csv	31-03-2020 16:28	Microsoft Excel Co...	1 KB

A4 MANOJ

	A	B
1	name	attendence
2	ADHI	0
3	MANJU	0
4	MANOJ	1
5	RAJIV	0
6	SHARAN	0

Fig. 6.13 File location and the attendance marked to excel sheet

The above images shows the path of the excel sheet where the attendance is marked and the excel sheet of marked attendance. For all the classes it will create a folder with the name as date in the attendance folder. At a particular class the folder is created for particular date folder, In that folder the attendance spread sheet is created for all class in respective folder.

The spread sheet there is a name and attendance rows if the student is present for that particular class attendance will be marked as 1 else it is marked as 0 for a Particular students.

CHAPTER 07

ADVANTAGES AND APPLICATIONS

ADVANTAGES:

- It is time saving as the image of the student is captured in the fraction of second and the attendance of the present students are marked without any manual confusion leading to save a lot of time and increases convenience of the teachers.
- It also avoids the chances of proxies by the friends of a student and duplication of one's identity by another. As no two students except twins will have same faces so the system cannot mark false attendance for a student who is not present in the class.
- The data is recorded by the computer and hence have high accuracy and there are no chances of human errors. The machine is very accurate and provides the business with error-free information of the students.
- The monitoring system updates the attendance on the spot and hence it will not be altered later by anyone. The updating of database and capturing of the student's images occurs simultaneously.
- It can store more than one image for a user to maximize the accuracy of face detection and less chances of error with updating the attendance database.
- As the students are not compelled to touch the monitor, they are only allowed to stand in front of the cameras as a result it defends the people from the physical illness like flu, virus and bacterial diseases.
- This system has massive data storage and all of that can be encrypted too, so there are very low chances of data theft and all that record can be stored for a long period of time. In case of manual attendance a teacher is required to change the attendance sheet or register which can be lost by human error and all that data can be lost.

APPLICATIONS:

- **INSTITUTIONS**

  Institutions have a traditional way of marking attendees at each student's name to check their attendance. This type of roll call is time-consuming and tedious. By using facial recognition, the attendance process can be greatly improved to save time and provide a convenient way to tag attendees. As the number of students at the institution is higher, using the automated system improves productivity and college quality.

- **COMPANIES**

  At most companies, employees have a habit of using their biometric or ID card to enter their check-in and check-out time. During peak hours the number of people entering and leaving the office is usually high. This causes a breakup in the workplace and people reach the line waiting for their turn. Face Recognition Systems provides an easy way to manage this attendance process. Employees do not have to worry about entering their time as an automated process. The system will monitor the time of entry and exit when the employee enters or exits the office.

- **PRISON**

  In prison, daily the head count of prisoners is carried out to check that all inmates are there. Facial recognition is used to activate the process of performing head counting to increase efficiency and trust. Security is also increasing as the tabs can be kept in each prisoner at all times.

- **SEMINARS / CONFERENCES**

  At conferences and international gatherings where large numbers of people gather, monitoring their presence is a daunting task when using biometrics or other means of monitoring. In that case using face recognition is easy, time-saving and very helpful. The camera can take pictures of all faces at once and will look at them.

CHAPTER 08

FUTURE SCOPE

- The automated Attendance System using Face detection can be set up in larger venues such as a conference hall where it helps to feel the presence of many people.

- Sometimes poor lighting conditions in the classroom can affect image quality that negatively affects the performance of the program, this can be overcome in the final stage by improving video quality or by using specific algorithms.

- In future the monitoring using face recognition can also be used at AVIATION so that people can avail paperless travel at airports. With improvement and advancement in the technology the boarding, checking-in, checking-out will be hassle free.

- In entertainment sector also it can be used like people can get access to multiplex cinema with just their image recorded at the entrance and there will not be any queues for the tickets near the counter. This will provide leisure and convenience to the future generation.

CHAPTER 9:
CONCLUSION

Thus, the aim of this project is to capture the images of the students, relate it with the database to ensure their presence or absence, mark attendance to the particular student to maintain the record. The Automated Classroom Attendance System helps in increasing the accuracy and speed, ultimately achieve the high-precision real-time attendance to meet the need for automatic classroom attendance.

A major factor in the development of the program is to remove all issues that were associated with attendance. Obstacles that arise from wasting time and time on paper, until representative issues from the classroom, will be completely eliminated.

Therefore, desirable results with a user-friendly interface are expected in the future, in the system. The effectiveness of the program can be increased by incorporating various steps and strategies into the developing stages of the program.

REFERENCES

- Rajath S Bharadwaj, Tejus S Rao, Vinay T R

  International Journal of Innovative Technology and Exploring Engineering (IJITEE)

  ISSN: 2278-3075, Volume-8 Issue-6, April 2019

  Will berger.org

- Lukas, S., Mitra, A. R., Desanti, R. I., & Krisnadi, D. (2016) Student attendance system in classroom using face recognition technique International Conference on Information and Communication Technology Convergence (ICTC). doi:10.1109/ictc.2016.7763360

Web reference

- Mathworks

  https://www.mathworks.com/

- Matlab Central

  https://www.mathworks.com/matlabcentral/

- Willberger.org

APPENDIX

MATLAB CODE:

```
classdef capture < matlab.apps.AppBase

  % Properties that correspond to app components
  properties (Access = public)
    UIFigure            matlab.ui.Figure               % Student...
    TabGroup            matlab.ui.container.TabGroup   % capture...
    Tab                 matlab.ui.container.Tab        % capture...
    LabelEditField      matlab.ui.control.Label        % ENTER S...
    EditField           matlab.ui.control.EditField
    Button              matlab.ui.control.Button       % TAKE PI...
    Label               matlab.ui.control.Label        % NEW HOR...
    Tab2                matlab.ui.container.Tab        % Attende...
    Label2              matlab.ui.control.Label        % NEW HOR...
    Panel3              matlab.ui.container.Panel      % Attendence
    Button5             matlab.ui.control.Button       % Take at...
    CheckBox            matlab.ui.control.CheckBox     % Stop
    Button6             matlab.ui.control.Button       % Train I...
    Tab3                matlab.ui.container.Tab        % Holidays
    Panel               matlab.ui.container.Panel      % HOLIDAY...
    LabelNumericEditField  matlab.ui.control.Label     % DAY
    NumericEditField    matlab.ui.control.NumericEditField % [-Inf Inf]
    LabelNumericEditField2 matlab.ui.control.Label     % MONTH
    NumericEditField2   matlab.ui.control.NumericEditField % [-Inf Inf]
    LabelNumericEditField3 matlab.ui.control.Label     % YEAR
    NumericEditField3   matlab.ui.control.NumericEditField % [-Inf Inf]
    Button3             matlab.ui.control.Button       % ENTER
    Button2             matlab.ui.control.Button       % CLEAR
    Label3              matlab.ui.control.Label        % NEW HOR...
    Button8             matlab.ui.control.Button       % Show ho...
    Tab4                matlab.ui.container.Tab        % Timings
    Label4              matlab.ui.control.Label        % NEW HOR...
    Panel2              matlab.ui.container.Panel      % Timings...
    LabelNumericEditField4 matlab.ui.control.Label     % Start time
    NumericEditField4   matlab.ui.control.NumericEditField % [-Inf Inf]
    LabelNumericEditField5 matlab.ui.control.Label     % End time
    NumericEditField5   matlab.ui.control.NumericEditField % [-Inf Inf]
    LabelNumericEditField6 matlab.ui.control.Label     % Lunch b...
    NumericEditField6   matlab.ui.control.NumericEditField % [-Inf Inf]
    LabelNumericEditField7 matlab.ui.control.Label     % Capture...
    NumericEditField7   matlab.ui.control.NumericEditField % [-Inf Inf]
    Button4             matlab.ui.control.Button       % ENTER
```

```
        Button7        matlab.ui.control.Button      % Show ti...
    end


    properties (Access = private)
        name; % Description
        da_y;
        mont_h;
        yea_r;
        stim_e;
        etim_e;
        ltim_e;
        ctim_e;
        bre;
    end




    methods (Access = private)

        % Code that executes after component creation
        function startupFcn(app)
            a=exist('date_.mat');
            if(a==0)
                d=0;
                m=0;
                y=0;
                save('date_.mat','d','m','y');
            end
            a=exist('time_.mat');
            if(a==0)
                s=9;
                e=16;
                b=13;
                c=3;
                save('time_.mat','s','e','b','c');
            end
            app.bre=0;
        end

        % EditField value changed function
        function EditFieldValueChanged(app)
            value = app.EditField.Value;
            app.name=value;

        end
```

```
% Button button pushed function
function ButtonButtonPushed(app)
   if(length(app.name)==0)
      fig = uifigure;
      uialert(fig,'enter the name please','error');
   else
      name_=app.name;
      nu_image=25;
      cam=webcam;
      image_s=zeros(200,200,30,'uint8');
      mm=1;
      for m=1:nu_image*2
         if(mm<=nu_image)
         img=snapshot(cam);
         subplot(2,1,1);
         imshow(img)
         FaceDetect=vision.CascadeObjectDetector;
         FaceDetect.MergeThreshold=5;
         bb=step(FaceDetect,img);
         hold on
         if(length(bb)>3)
            rectangle('Position',bb(1,:),'LineWidth',3)
            hold off
            img=imcrop(img,bb(1,:));
            i_1=rgb2gray(img);
           i_1 =imresize(i_1,[200,200]);
            image_s(:,:,mm)=i_1;
            subplot(2,1,2);
            imshow(image_s(:,:,mm));
            mm=mm+1;
         end
         pause(0.1);
         end
      end
      clear cam
      if(mm==nu_image+1)
         fold='C:\Users\MANOJ N\Desktop\test2';
          mkdir(fold,name_);
         for m=1:nu_image
            name_1=strcat(fold,'\',name_,'\',name_,int2str(m),'.jpg');
            imwrite(image_s(:,:,m),name_1);
         end
      else
         fig = uifigure;
         uialert(fig,'please show face','error');
      end
 end
```

```
end

% NumericEditField value changed function
function NumericEditFieldValueChanged(app)
   value = app.NumericEditField.Value;
  app.da_y=value;
end

% NumericEditField3 value changed function
function NumericEditField3ValueChanged(app)
   value = app.NumericEditField3.Value;
   app.yea_r=value;
end

% NumericEditField2 value changed function
function NumericEditField2ValueChanged(app)
   value = app.NumericEditField2.Value;
   app.mont_h=value;

end

% Button3 button pushed function
function Button3ButtonPushed(app)
  a= exist('date_.mat');
  if(a==0)
     d=app.da_y;
     m=app.mont_h;
     y=app.yea_r;
     save('date_.mat','d','m','y');
  else
   load('date_.mat');
      if(length(d)>=1)
       d(end+1)=app.da_y;
       m(end+1)=app.mont_h;
       y(end+1)=app.yea_r;
      else
       d=app.da_y;
       m=app.mont_h;
       y=app.yea_r;
      end
   save('date_.mat','d','m','y');
  end
end

% Button2 button pushed function
function Button2ButtonPushed(app)
   delete('date_.mat');
end
```

```
% NumericEditField4 value changed function
function NumericEditField4ValueChanged(app)
   value = app.NumericEditField4.Value;
    app.stim_e=value;
end

% NumericEditField5 value changed function
function NumericEditField5ValueChanged(app)
   value = app.NumericEditField5.Value;
   app.etim_e=value;
end

% NumericEditField6 value changed function
function NumericEditField6ValueChanged(app)
   value = app.NumericEditField6.Value;
   app.ltim_e=value;
end

% NumericEditField7 value changed function
function NumericEditField7ValueChanged(app)
   value = app.NumericEditField7.Value;
   app.ctim_e=value;
end

% Button4 button pushed function
function Button4ButtonPushed(app)
      s=app.stim_e;
      e=app.etim_e;
      b=app.ltim_e;
      c=app.ctim_e;
      save('time_.mat','s','e','b','c');
end

% Button5 button pushed function
function Button5ButtonPushed(app)

    if (exist('train.mat')==0) % first if
   fig = uifigure;
   uialert(fig,'first train','error');

else
     load('train.mat');
     load('time_.mat');
     load('date_.mat');
     collection=imageSet('C:\Users\MANOJ N\Desktop\test2','recursive');
        while 1 %infinet
          clk=clock;
```

```
day_=day(datetime(clk(1),clk(2),clk(3)),'name');
ans_=strcmp(day_,'Sunday');
ans1_=prod(clk(1)~=y(:))+prod(clk(2)~=m(:))+prod(clk(3)~=d(:));
if(app.bre==1)
   break;
else
   if(ans_==0 && ans1_~=0) %not working day
      clk=clock;
         if((clk(4)>=s && clk(4)<=e) && (app.bre~=1)) %working day
           clk=clock;
           folder1=strcat(int2str(clk(1)),'-',int2str(clk(2)),'-',int2str(clk(3)));
           file_name1='C:\Users\MANOJ N\Downloads\attendance';
           mkdir(file_name1,folder1);
           while 1 %working day while
              clk=clock;
              if((~((clk(4)>=s && clk(4)<=e )) || (app.bre==1))) %working day while breaking if
                 break

              else
                 if((clk(5)<=c && (clk(4)~=b) ) && (app.bre~=1)) %working time
                    file_name2=strcat(file_name1,'\',folder1);
                    folder2=strcat(int2str(clk(4)));
                    mkdir(file_name2,folder2);
                    cam=webcam;
                    val=zeros(1,length(collection));
                    tab=table(name_id',val','VAriableNames',{'name' 'attendence'});
                    while 1 %working time while
                       clk=clock;
                       if((~(((clk(5)<=c) && (clk(4)~=b))) || (app.bre==1) ))%working time while breaking if
                          break

                       else
                           pause(1);
                            FaceDetect=vision.CascadeObjectDetector;
                            FaceDetect.MergeThreshold=5;

                              img=snapshot(cam);
                              subplot(2,2,1);
                              imshow(img)

                              bb=step(FaceDetect,img);
                              subplot(2,2,2);
                              imshow(img);
                              hold on

                              if(length(bb)>3)
                                 rectangle('Position',bb(1,:),'LineWidth',3)
                              hold off
                    img=imcrop(img,bb(1,:));
                    subplot(2,2,3);
                    imshow(img);
                    image_capture=rgb2gray(img);
                    image_capture=imresize(image_capture,[200,200]);
                    subplot(2,2,4);
                    imshow(image_capture);
                    features123=extractHOGFeatures(image_capture);
                    na_me=predict(class1,features123);
                    for i=1:length(collection)
                      if(strcmp(collection(i).Description,na_me)==1)
                      name_nu=i;
                      end
                    end
                    m=extractHOGFeatures(image_capture);
                    for i=1:5
                    n=extractHOGFeatures(read(collection(name_nu),i));
                    l=m-n;
                    l=l.*l;
                    num(i)=predict(class_,l);
                    end
                     if(sum(num)>3)
                      present=collection(name_nu).Description;
                      tab{name_nu,2}=1;

                     end
                    end

                end %working time while breaking if end
                pause(0.3);
              end %working time while end
               clear cam
               file_name3=strcat(file_name2,'\',folder2,'\','attendence.csv');
               writetable(tab,file_name3);
               display('interupted1');

            end %working time end
            pause(1);
            display('interupted2');
          end %working day while breaking if end
        end %working day while end
        pause(1);
        display('interupted3');
      end %working day
      display('interupted4');
  end %not working day
  display('interupted5');
end%end infinet loop if
```

```
                pause(1);
                display('interupted6');
              end %end  infinet while
 display('interupted7');
end %first if
display('attendence stoped');
        end

        % CheckBox value changed function
        function CheckBoxValueChanged(app)
            value = app.CheckBox.Value;
            app.bre=value;
        end

        % Button6 button pushed function
        function Button6ButtonPushed(app)
            collection=imageSet('C:\Users\MANOJ N\Desktop\test2','recursive');
            a=1;
            features = zeros(length(collection)*collection(1).Count,20736);
            for i=1:length(collection)
                for j = 1:collection(1).Count
                    features(a , :) = extractHOGFeatures(read(collection(i),j));
                    names_{a}=collection(i).Description;
                    a = a + 1;
                end
                name_id{i} = collection(i).Description;
            end
            numb= zeros(60,20736);
            a=1;
            for i=1:3
                for j=1:10
                    m=extractHOGFeatures(read(collection(i),j));
                    n=extractHOGFeatures(read(collection(i),j+4));
                    l=m-n;
                    l=l.*l;
                    numb(a , :)=l;
                    pre(a)=1;
                    a=a+1;
                end
            end
            for i=1:3
                for j=1:10
                    m=extractHOGFeatures(read(collection(i),j));
                    n=extractHOGFeatures(read(collection(i+1),j+4));
                    l=m-n;
                    l=l.*l;
                    numb(a , :) =l;
                    pre(a)=0;
```

```
          a=a+1;
        end
      end


       class1= fitcecoc(features,names_);
       class_= fitcecoc(numb,pre);
       save('train.mat','class1','class_','name_id');
       display('training done')

    end

    % Button7 button pushed function
    function Button7ButtonPushed(app)
      display(load('time_.mat'));
    end

    % Button8 button pushed function
    function Button8ButtonPushed(app)
      display(load('date_.mat'));

    end
  end

  % App initialization and construction
  methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

      % Create UIFigure
      app.UIFigure = uifigure;
      app.UIFigure.Position = [100 100 757 481];
      app.UIFigure.Name = 'Students attendence monetering system';
      setAutoResize(app, app.UIFigure, true)

      % Create TabGroup
      app.TabGroup = uitabgroup(app.UIFigure);
      app.TabGroup.TabLocation = 'right';
      app.TabGroup.Units = 'pixels';
      app.TabGroup.Position = [18 12 726 458];

      % Create Tab
      app.Tab = uitab(app.TabGroup);
      app.Tab.Units = 'pixels';
      app.Tab.Title = 'capture image';
      app.Tab.BackgroundColor = [0.6 0.9686 0.6745];
```

```
% Create LabelEditField
app.LabelEditField = uilabel(app.Tab);
app.LabelEditField.HorizontalAlignment = 'center';
app.LabelEditField.VerticalAlignment = 'center';
app.LabelEditField.FontName = 'Cambria Math';
app.LabelEditField.FontSize = 24;
app.LabelEditField.FontWeight = 'bold';
app.LabelEditField.FontColor = [0.8863 0.4275 0.1255];
app.LabelEditField.Position = [93 337 258 30];
app.LabelEditField.Text = 'ENTER STUDENT NAME';

% Create EditField
app.EditField = uieditfield(app.Tab, 'text');
app.EditField.ValueChangedFcn = createCallbackFcn(app, @EditFieldValueChanged);
app.EditField.HorizontalAlignment = 'center';
app.EditField.FontName = 'Bookman';
app.EditField.FontSize = 24;
app.EditField.FontWeight = 'bold';
app.EditField.FontColor = [0.8588 0.5333 0.3176];
app.EditField.Position = [143 287 139.03125 40];

% Create Button
app.Button = uibutton(app.Tab, 'push');
app.Button.ButtonPushedFcn = createCallbackFcn(app, @ButtonButtonPushed);
app.Button.FontSize = 22;
app.Button.FontWeight = 'bold';
app.Button.Position = [130 185 152 43];
app.Button.Text = 'TAKE PICTURE';

% Create Label
app.Label = uilabel(app.Tab);
app.Label.FontSize = 24;
app.Label.FontWeight = 'bold';
app.Label.FontColor = [1 0.498 0];
app.Label.Position = [30 417 419 31];
app.Label.Text = 'NEW HORIZON COLLEGE OF ENGINEERING';

% Create Tab2
app.Tab2 = uitab(app.TabGroup);
app.Tab2.Units = 'pixels';
app.Tab2.Title = 'Attendence ';
app.Tab2.BackgroundColor = [0.8392 0.8392 0.3451];

% Create Label2
app.Label2 = uilabel(app.Tab2);
app.Label2.FontSize = 24;
app.Label2.FontWeight = 'bold';
app.Label2.FontColor = [1 0.498 0];
app.Label2.Position = [30 417 419 31];
app.Label2.Text = 'NEW HORIZON COLLEGE OF ENGINEERING';

% Create Panel3
app.Panel3 = uipanel(app.Tab2);
app.Panel3.BorderType = 'line';
app.Panel3.Title = 'Attendence';
app.Panel3.FontName = 'Helvetica';
app.Panel3.FontUnits = 'pixels';
app.Panel3.FontSize = 12;
app.Panel3.Units = 'pixels';
app.Panel3.Position = [129 177 222 221];

% Create Button5
app.Button5 = uibutton(app.Panel3, 'push');
app.Button5.ButtonPushedFcn = createCallbackFcn(app, @Button5ButtonPushed);
app.Button5.FontName = 'Lucida Sans';
app.Button5.FontSize = 20;
app.Button5.Position = [37.5 148 129 36];
app.Button5.Text = 'Take attendence';

% Create CheckBox
app.CheckBox = uicheckbox(app.Panel3);
app.CheckBox.ValueChangedFcn = createCallbackFcn(app, @CheckBoxValueChanged);
app.CheckBox.Text = 'Stop ';
app.CheckBox.FontName = 'Ebrima';
app.CheckBox.FontSize = 20;
app.CheckBox.Position = [77 74 65.359375 27];

% Create Button6
app.Button6 = uibutton(app.Tab2, 'push');
app.Button6.ButtonPushedFcn = createCallbackFcn(app, @Button6ButtonPushed);
app.Button6.FontSize = 16;
app.Button6.Position = [183 111 100 26];
app.Button6.Text = 'Train Images';

% Create Tab3
app.Tab3 = uitab(app.TabGroup);
app.Tab3.Units = 'pixels';
app.Tab3.Title = 'Holidays';
app.Tab3.BackgroundColor = [0.2941 0.5882 0.7843];

% Create Panel
app.Panel = uipanel(app.Tab3);
app.Panel.BorderType = 'line';
app.Panel.Title = 'HOLIDAY ENTERY';
app.Panel.BackgroundColor = [0.2941 0.5882 0.7843];
app.Panel.FontName = 'Helvetica';
app.Panel.FontUnits = 'pixels';
app.Panel.FontSize = 12;
app.Panel.Units = 'pixels';
app.Panel.Position = [99 124 256 265];

% Create LabelNumericEditField
app.LabelNumericEditField = uilabel(app.Panel);
app.LabelNumericEditField.HorizontalAlignment = 'right';
app.LabelNumericEditField.VerticalAlignment = 'bottom';
app.LabelNumericEditField.Position = [53.03125 184 21 15];
app.LabelNumericEditField.Text = 'DAY';

% Create NumericEditField
app.NumericEditField = uieditfield(app.Panel, 'numeric');
app.NumericEditField.ValueChangedFcn = createCallbackFcn(app, @NumericEditFieldValueChanged);
app.NumericEditField.Position = [89.03125 180 100 22];

% Create LabelNumericEditField2
app.LabelNumericEditField2 = uilabel(app.Panel);
app.LabelNumericEditField2.HorizontalAlignment = 'right';
app.LabelNumericEditField2.Position = [45.03125 138 37 15];
app.LabelNumericEditField2.Text = 'MONTH';

% Create NumericEditField2
app.NumericEditField2 = uieditfield(app.Panel, 'numeric');
app.NumericEditField2.ValueChangedFcn = createCallbackFcn(app, @NumericEditField2ValueChanged);
app.NumericEditField2.Position = [89.03125 134 100 22];

% Create LabelNumericEditField3
app.LabelNumericEditField3 = uilabel(app.Panel);
app.LabelNumericEditField3.HorizontalAlignment = 'center';
app.LabelNumericEditField3.Position = [45 99 37.03125 15];
app.LabelNumericEditField3.Text = 'YEAR';

% Create NumericEditField3
app.NumericEditField3 = uieditfield(app.Panel, 'numeric');
app.NumericEditField3.ValueChangedFcn = createCallbackFcn(app, @NumericEditField3ValueChanged);
app.NumericEditField3.Position = [89.03125 92 100 22];

% Create Button3
app.Button3 = uibutton(app.Panel, 'push');
app.Button3.ButtonPushedFcn = createCallbackFcn(app, @Button3ButtonPushed);
app.Button3.Position = [89 30 100 22];
app.Button3.Text = 'ENTER';

% Create Button2
app.Button2 = uibutton(app.Tab3, 'push');
app.Button2.ButtonPushedFcn = createCallbackFcn(app, @Button2ButtonPushed);
app.Button2.FontSize = 16;
app.Button2.FontWeight = 'bold';
app.Button2.Position = [190 75 100 26];
app.Button2.Text = 'CLEAR';

% Create Label3
app.Label3 = uilabel(app.Tab3);
app.Label3.FontSize = 24;
app.Label3.FontWeight = 'bold';
app.Label3.FontColor = [1 0.498 0];
app.Label3.Position = [30 417 419 31];
app.Label3.Text = 'NEW HORIZON COLLEGE OF ENGINEERING';

% Create Button8
app.Button8 = uibutton(app.Tab3, 'push');
app.Button8.ButtonPushedFcn = createCallbackFcn(app, @Button8ButtonPushed);
app.Button8.Position = [403 302 100 22];
app.Button8.Text = 'Show holidays';

% Create Tab4
app.Tab4 = uitab(app.TabGroup);
app.Tab4.Units = 'pixels';
app.Tab4.Title = 'Timings';
app.Tab4.BackgroundColor = [0.7255 0.5686 0.451];

% Create Label4
app.Label4 = uilabel(app.Tab4);
app.Label4.FontSize = 24;
app.Label4.FontWeight = 'bold';
app.Label4.FontColor = [1 0.498 0];
app.Label4.Position = [30 417 419 31];
app.Label4.Text = 'NEW HORIZON COLLEGE OF ENGINEERING';

% Create Panel2
app.Panel2 = uipanel(app.Tab4);
app.Panel2.BorderType = 'line';
app.Panel2.Title = 'Timings entery';
app.Panel2.BackgroundColor = [0.7255 0.5686 0.451];
app.Panel2.FontName = 'Helvetica';
app.Panel2.FontUnits = 'pixels';
app.Panel2.FontSize = 12;
app.Panel2.Units = 'pixels';
app.Panel2.Position = [124 148 231 252];

% Create LabelNumericEditField4
app.LabelNumericEditField4 = uilabel(app.Panel2);
app.LabelNumericEditField4.HorizontalAlignment = 'right';
app.LabelNumericEditField4.Position = [19.03125 202 43 15];
app.LabelNumericEditField4.Text = 'Start time';

% Create NumericEditField4
app.NumericEditField4 = uieditfield(app.Panel2, 'numeric');
app.NumericEditField4.ValueChangedFcn = createCallbackFcn(app, @NumericEditField4ValueChanged);
app.NumericEditField4.Position = [75.03125 198 100 22];

% Create LabelNumericEditField5
app.LabelNumericEditField5 = uilabel(app.Panel2);
app.LabelNumericEditField5.HorizontalAlignment = 'right';
app.LabelNumericEditField5.Position = [21.03125 153 39 15];
app.LabelNumericEditField5.Text = 'End time';

% Create NumericEditField5
app.NumericEditField5 = uieditfield(app.Panel2, 'numeric');
app.NumericEditField5.ValueChangedFcn = createCallbackFcn(app, @NumericEditField5ValueChanged);
app.NumericEditField5.Position = [75.03125 149 100 22];

% Create LabelNumericEditField6
app.LabelNumericEditField6 = uilabel(app.Panel2);
app.LabelNumericEditField6.HorizontalAlignment = 'right';
app.LabelNumericEditField6.Position = [5.03125 103 55 15];
app.LabelNumericEditField6.Text = 'Lunch break';

% Create NumericEditField6
app.NumericEditField6 = uieditfield(app.Panel2, 'numeric');
app.NumericEditField6.ValueChangedFcn = createCallbackFcn(app, @NumericEditField6ValueChanged);
app.NumericEditField6.Position = [75.03125 99 100 22];

% Create LabelNumericEditField7
app.LabelNumericEditField7 = uilabel(app.Panel2);
app.LabelNumericEditField7.HorizontalAlignment = 'right';
app.LabelNumericEditField7.Position = [5.03125 54 57 15];
app.LabelNumericEditField7.Text = 'Capture time';

% Create NumericEditField7
app.NumericEditField7 = uieditfield(app.Panel2, 'numeric');
app.NumericEditField7.ValueChangedFcn = createCallbackFcn(app, @NumericEditField7ValueChanged);
app.NumericEditField7.Position = [77.03125 50 100 22];

% Create Button4
app.Button4 = uibutton(app.Panel2, 'push');
app.Button4.ButtonPushedFcn = createCallbackFcn(app, @Button4ButtonPushed);
app.Button4.Position = [77 10 100 22];
app.Button4.Text = 'ENTER';

% Create Button7
app.Button7 = uibutton(app.Tab4, 'push');
```

```
      app.Button7.ButtonPushedFcn = createCallbackFcn(app, @Button7ButtonPushed);
      app.Button7.Position = [390 316 100 22];
      app.Button7.Text = 'Show timings';
    end
  end

  methods (Access = public)

    % Construct app
    function app = capture()

      % Create and configure components
      createComponents(app)

      % Register the app with App Designer
      registerApp(app, app.UIFigure)

      % Execute the startup function
      runStartupFcn(app, @startupFcn)

      if nargout == 0
        clear app
      end
    end

    % Code that executes before app deletion
    function delete(app)

      % Delete UIFigure when app is deleted
      delete(app.UIFigure)
    end
  end
end
```

PYTHON CODE:

```
                    import tkinter as tk
from tkinter import Message ,Text
import cv2,os
import shutil
import csv
import numpy as np
from PIL import Image, ImageTk
import pandas as pd
import datetime
import time
import tkinter.ttk as ttk
import tkinter.font as font

window = tk.Tk()
#helv36 = tk.Font(family='Helvetica', size=36, weight='bold')
window.title("Face_Recogniser")

dialog_title = 'QUIT'
dialog_text = 'Are you sure?'
#answer = messagebox.askquestion(dialog_title, dialog_text)

window.geometry('1280x720')
window.configure(background='navy')

#window.attributes('-fullscreen', True)

window.grid_rowconfigure(0, weight=1)
window.grid_columnconfigure(0, weight=1)

#path = "profile.jpg"

#Creates a Tkinter-compatible photo image, which can be used everywhere Tkinter expects an image object.
#img = ImageTk.PhotoImage(Image.open(path))

#The Label widget is a standard Tkinter widget used to display a text or image on the screen.
#panel = tk.Label(window, image = img)


#panel.pack(side = "left", fill = "y", expand = "no")

#cv_img = cv2.imread("img541.jpg")
#x, y, no_channels = cv_img.shape
#canvas = tk.Canvas(window, width = x, height =y)
#canvas.pack(side="left")
```

```
#photo = PIL.ImageTk.PhotoImage(image = PIL.Image.fromarray(cv_img))
# Add a PhotoImage to the Canvas
#canvas.create_image(0, 0, image=photo, anchor=tk.NW)

#msg = Message(window, text='Hello, world!')

# Font is a tuple of (font_family, size_in_points, style_modifier_string)



# message = tk.Label(window, text="Face-Recognition-Based-Attendance-Management-System" ,bg="Green"
,fg="white"  ,width=50  ,height=3,font=('times', 30, 'italic bold underline'))

# message.place(x=200, y=20)

lbl = tk.Label(window, text="Enter USN",width=20  ,height=2  ,fg="black"  ,bg="white" ,font=('times', 15, ' bold ') )
lbl.place(x=100, y=100)

txt = tk.Entry(window,width=20 ,bg="white" ,fg="black",font=('times', 15, ' bold '))
txt.place(x=550, y=105)

lbl2 = tk.Label(window, text="Enter Name",width=20  ,fg="black"  ,bg="white"    ,height=2 ,font=('times', 15, ' bold
'))
lbl2.place(x=100, y=200)

txt2 = tk.Entry(window,width=20  ,bg="white"  ,fg="black",font=('times', 15, ' bold ')  )
txt2.place(x=550, y=205)

lbl3 = tk.Label(window, text="Notification:",width=20  ,fg="black"  ,bg="white"  ,height=2 ,font=('times', 15, ' bold
underline '))
lbl3.place(x=100, y=300)

message = tk.Label(window, text="" ,bg="white"  ,fg="black"  ,width=30  ,height=2, activebackground = "white"
,font=('times', 15, ' bold '))
message.place(x=500, y=305)

lbl3 = tk.Label(window, text="Attendance:",width=20  ,fg="black"  ,bg="white"  ,height=2 ,font=('times', 15, ' bold
underline'))
lbl3.place(x=400, y=550)


message2 = tk.Label(window, text="" ,fg="black"   ,bg="white",activeforeground = "white",width=30  ,height=2
,font=('times', 15, ' bold '))
message2.place(x=700, y=550)

def clear():
   txt.delete(0, 'end')
   res = ""
```

```
    message.configure(text= res)

def clear2():
    txt2.delete(0, 'end')
    res = ""
    message.configure(text= res)

def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        pass

    try:
        import unicodedata
        unicodedata.numeric(s)
        return True
    except (TypeError, ValueError):
        pass

    return False

def TakeImages():
    Id=(txt.get())
    name=(txt2.get())
    if(is_number(Id) and name.isalpha()):
        cam = cv2.VideoCapture(0)
        harcascadePath = "haarcascade_frontalface_default.xml"
        detector=cv2.CascadeClassifier(harcascadePath)
        sampleNum=0
        while(True):
            ret, img = cam.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = detector.detectMultiScale(gray, 1.3, 5)
            for (x,y,w,h) in faces:
                cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
                #incrementing sample number
                sampleNum=sampleNum+1
                #saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage\ "+name +"."+Id +'.'+ str(sampleNum) + ".jpg", gray[y:y+h,x:x+w])
                #display the frame
                cv2.imshow('frame',img)
            #wait for 100 miliseconds
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            # break if the sample number is morethan 100
            elif sampleNum>60:
```

```
            break
        cam.release()
        cv2.destroyAllWindows()
        res = "Images Saved for ID : " + Id +" Name : "+ name
        row = [Id , name]
        with open('StudentDetails\StudentDetails.csv','a+') as csvFile:
            writer = csv.writer(csvFile)
            writer.writerow(row)
        csvFile.close()
        message.configure(text= res)
    else:
        if(is_number(Id)):
            res = "Enter Alphabetical Name"
            message.configure(text= res)
        if(name.isalpha()):
            res = "Enter Numeric Id"
            message.configure(text= res)

def TrainImages():
    recognizer = cv2.face_LBPHFaceRecognizer.create()#recognizer = 
cv2.face.LBPHFaceRecognizer_create()#$cv2.createLBPHFaceRecognizer()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector =cv2.CascadeClassifier(harcascadePath)
    faces,Id = getImagesAndLabels("TrainingImage")
    recognizer.train(faces, np.array(Id))
    recognizer.save("TrainingImageLabel\Trainner.yml")
    res = "Image Trained"#+",".join(str(f) for f in Id)
    message.configure(text= res)

def getImagesAndLabels(path):
    #get the path of all the files in the folder
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
    #print(imagePaths)

    #create empth face list
    faces=[]
    #create empty ID list
    Ids=[]
    #now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        #loading the image and converting it to gray scale
        pilImage=Image.open(imagePath).convert('L')
        #Now we are converting the PIL image into numpy array
        imageNp=np.array(pilImage,'uint8')
        #getting the Id from the image
        Id=int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        Ids.append(Id)
    return faces,Ids

def TrackImages():
    recognizer = cv2.face.LBPHFaceRecognizer_create()#cv2.createLBPHFaceRecognizer()
    recognizer.read("TrainingImageLabel\Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);
    df=pd.read_csv("StudentDetails\StudentDetails.csv")
    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names =  ['Id','Name','Date','Time']
    attendance = pd.DataFrame(columns = col_names)
    while True:
        ret, im =cam.read()
        gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
        faces=faceCascade.detectMultiScale(gray, 1.2,5)
        for(x,y,w,h) in faces:
            cv2.rectangle(im,(x,y),(x+w,y+h),(225,0,0),2)
            Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
            if(conf < 50):
                ts = time.time()
                date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
                timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                aa=df.loc[df['Id'] == Id]['Name'].values
                tt=str(Id)+"-"+aa
                attendance.loc[len(attendance)] = [Id,aa,date,timeStamp]

            else:
                Id='Unknown'
                tt=str(Id)
            if(conf > 75):
                noOfFile=len(os.listdir("ImagesUnknown"))+1
                cv2.imwrite("ImagesUnknown\Image"+str(noOfFile) + ".jpg", im[y:y+h,x:x+w])
            cv2.putText(im,str(tt),(x,y+h), font, 1,(255,255,255),2)
        attendance=attendance.drop_duplicates(subset=['Id'],keep='first')
        cv2.imshow('im',im)
        if (cv2.waitKey(1)==ord('q')):
            break
    ts = time.time()
    date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
    timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
    Hour,Minute,Second=timeStamp.split(":")
    fileName="Attendance\Attendance_"+date+"_"+Hour+"-"+Minute+"-"+Second+".csv"
    attendance.to_csv(fileName,index=False)
    cam.release()
    cv2.destroyAllWindows()
    #print(attendance)
    res=attendance
    message2.configure(text= res)


clearButton = tk.Button(window, text="Clear", command=clear  ,fg="black"  ,bg="white"  ,width=20  ,height=2
,activebackground = "Red" ,font=('times', 15, ' bold '))
clearButton.place(x=950, y=100)
clearButton2 = tk.Button(window, text="Clear", command=clear2  ,fg="black"  ,bg="white"  ,width=20  ,height=2,
activebackground = "Red" ,font=('times', 15, ' bold '))
clearButton2.place(x=950, y=200)
takeImg = tk.Button(window, text="Take Images", command=TakeImages  ,fg="black"  ,bg="white"  ,width=20
,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))
takeImg.place(x=150, y=400)
trainImg = tk.Button(window, text="Train Images", command=TrainImages  ,fg="black"  ,bg="white"  ,width=20
,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))
trainImg.place(x=450, y=400)
trackImg = tk.Button(window, text="Track Images", command=TrackImages  ,fg="black"  ,bg="white"  ,width=20
,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))
trackImg.place(x=750, y=400)
quitWindow = tk.Button(window, text="Quit", command=window.destroy  ,fg="black"  ,bg="white"  ,width=20
,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))
quitWindow.place(x=1050, y=400)
copyWrite = tk.Text(window, background=window.cget("background"), borderwidth=0,font=('times', 30, 'italic
bold underline'))
copyWrite.tag_configure("superscript", offset=10)
copyWrite.insert("insert", "Developed by Sharan")
copyWrite.configure(state="disabled",fg="black"  )
copyWrite.pack(side="left")
copyWrite.place(x=800, y=750)

window.mainloop()
```