SIX WEEKS INDUSTRIAL TRAINING

ON

FACE RECOGNITION ATTENDANCE SYSTEM

AT

CETPA INFOTECH PVT. LTD.,(NOIDA)

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By:

MADHAV SHARMA

11700188

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DAV UNIVERSITY, SARMASTPUR,

JALANDHAR – 144012

CERTIFICATES

CETPA INFOTECH PRIVATE LIMITED

(An ISO 9001:2015 Certified Company)

Certificate of Training

This is to certify that

MADHAV SHARMA

has successfully completed Six Weeks Summer Training on
"Machine Learning"
from 5 June to 16 July, 2019
at CETPA INFOTECH PVT. LTD., Noida.

Anil Kumar Singh
Director-Training

CETPA®
Because Knowledge Matters
www.cetpainfotech.com

Vikas Kalra
Director

Verify this Certificate by Registration Number ESMach5620196W4472263 at http://www.cetpainfotech.com

CETPA INFOTECH PRIVATE LIMITED

(An ISO 9001:2015 Certified Company)

Certificate of Training

This is to certify that
MADHAV SHARMA

has successfully completed Six Weeks Summer Training on
"Python"
from 5 June to 16 July, 2019
at CETPA INFOTECH PVT. LTD., Noida.

Anil Kumar Singh
Director-Training

CETPA®
Because Knowledge Matters
www.cetpainfotech.com

Vikas Kalra
Director

Verify this Certificate by Registration Number ESPyth5620196W4237301 at http://www.cetpainfotech.com

MICROSOFT • CERTIFICATE OF

Achievement

This certificate accredits that

Madhav Sharma

Microsoft Partner
Silver Learning

has successfully completed the following
Microsoft Official Course

Fundamental Of Machine Learning

MICROSOFT • CERTIFICATE OF ACHIEVEMENT •

Satya Nadella
Chief Executive Officer

Anil Singh
Microsoft Certified Trainer

Certificate of Honor

CETPA CERTIFIED
ISO 9001:2015 Certified

In honor of the outstanding performance and dedication we gladly Present

Certificate of Honor

to MR. MADHAV SHARMA

This certificate is for recognition of his/her highest standard of excellence in PYTHON & MACHINE LEARNING *held at CETPA Infotech Pvt. Ltd.*

Anil Kumar Singh
Director

CETPA®
TRAINING | DEVELOPMENT | PLACEMENT
www.cetpainfotech.com

Vikas Kalra
Director

DECLARATION

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I hereby declare that the Industrial Training Report during the period from 6-6-2019 to 15-7-2019 for the award of degree of B.Tech. in Computer Science and Engineering, DAV University, Sarmastpur, Jalandhar.

Date: 15/07/2019

Madhav Sharma

11700188

(E115A19)

ACKNOWLEDGMENT

I express my gratitude to all those who helped us in various stages of the development of this project. First, I would like to express my sincere gratitude indebtedness to Dr. Deshbandhu Gupta (Vice-Chancellor), Dr. Jasbir Rishi (Dean Academics) and Er. Nishi (Coordinator, Department of Computer Science and Engineering) DAV University for allowing me to undergo the Six weeks industrial training at **CETPA**. I am also thankful to all faculty members of Department of Computer Science and Engineering, for their true help, inspiration and for helping me for the preparation of the final report and presentation. Lastly, I pay my sincere thanks and gratitude to all the Staff Members of CETPA for their support and for making our training valuable and fruitful.

FACE RECOGNITION ATTENDANCE SYSTEM

A System For Everyone

ABSTRACT

Uniqueness or individuality of an individual face is the representation of one's identity. In this project face of an individual is used for the purpose of attendance making automatically. Attendance of the student is very important for every college, universities and school. Conventional methodology for taking attendance is by calling the name or roll number of the student and the attendance is recorded. Time consumption for this purpose is an important point of concern. Assume that the duration for one subject is around 60 minutes or 1 hour & to record attendance takes 5 to 10 minutes. For every tutor this is consumption of time. To stay away from these losses, an automatic process is used in this project which is based on image processing. In this project face detection and face recognition is used. Face detection is used to locate the position of face region and face recognition is used for marking the understudy's attendance. The database of all the students in the class is stored and when the face of the individual student matches with one of the faces stored in the database then the attendance is recorded.

Table of Contents

Title **Page no.**

Table of Figures

Chapter 1

Introduction

1.1. Introduction

Attendance is prime important for both the teacher and student of an educational organization. So it is very important to keep record of the attendance. The problem arises when we think about the traditional process of taking attendance in class room.
Calling name or roll number of the student for attendance is not only a problem of time consumption but also it needs energy. So an automatic attendance system can solve all above problems.

There are some automatic attendances making system which are currently used by much institution. One of such system is biometric technique and RFID system. Although it is automatic and a step ahead of traditional method it fails to meet the time constraint. The student has to wait in queue for giving attendance, which is time taking.

This project introduces an involuntary attendance marking system, devoid of any kind of interference with the normal teaching procedure. The system can be also implemented during exam sessions or in other teaching activities where attendance is highly essential. This system eliminates classical student identification such as calling name of the student, or checking respective identification cards of the student, which can not only interfere with the ongoing teaching process, but also can be stressful for students during examination sessions. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

1.2. Background

Face recognition is crucial in daily life in order to identify family, friends or someone we are familiar with. We might not perceive that several steps have actually taken in order to identify human faces. Human intelligence allows us to receive information and interpret the information in the recognition process. We receive information through the image projected into our eyes, by specifically retina in the form of light. Light is a form of electromagnetic waves which are radiated from a source onto an object and projected to human vision. After visual processing done by the human visual system, we actually classify shape, size, contour and the texture of the object in order to analyze the information. The analyzed information will be compared to other representations of objects or face that exist in our memory to recognize. In fact, it is a hard challenge to build an automated system to have the same capability as a human to recognize faces. However, we need large memory to recognize different faces, for example, in the Universities, there are a lot of students with different race and gender, it is impossible to remember every face of the individual without making mistakes. In order to overcome human limitations, computers with almost limitless memory, high processing speed and power are used in face recognition systems.

The human face is a unique representation of individual identity. Thus, face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time capture image with stored images in the database of that person.

Nowadays, face recognition system is prevalent due to its simplicity and awesome performance. For instance, airport protection systems and FBI use face recognition for criminal investigations by tracking

suspects, missing children and drug activities. Apart from that, Facebook which is a popular social networking website implement face recognition to allow the users to tag their friends in the photo for entertainment purposes (Sidney Fussell, 2018). Furthermore, Intel Company allows the users to use face recognition to get access to their online account. Apple allows the users to unlock their mobile phone, iPhone X by using face recognition.

The work on face recognition began in 1960. Woody Bledsoe, Helen Chan Wolf and Charles Bisson had introduced a system which required the administrator to locate eyes, ears, nose and mouth from images. The distance and ratios between the located features and the common reference points are then calculated and compared. The studies are further enhanced by Goldstein, Harmon, and Lesk in 1970 by using other features such as hair colour and lip thickness to automate the recognition. In 1988, Kirby and Sirovich first suggested principle component analysis (PCA) to solve face recognition problem. Many studies on face recognition

were then conducted continuously until today.

1.3. Problem Statement

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.

Difficulties of facial identification is the identification between known and unknown images.

- training process for face recognition student attendance system is slow and time-consuming.
- different lighting and head poses are often the problems that could degrade the performance of face recognition based student attendance system.

Hence, there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will be the evaluation points of the performance.

1.4. Aims and Objectives

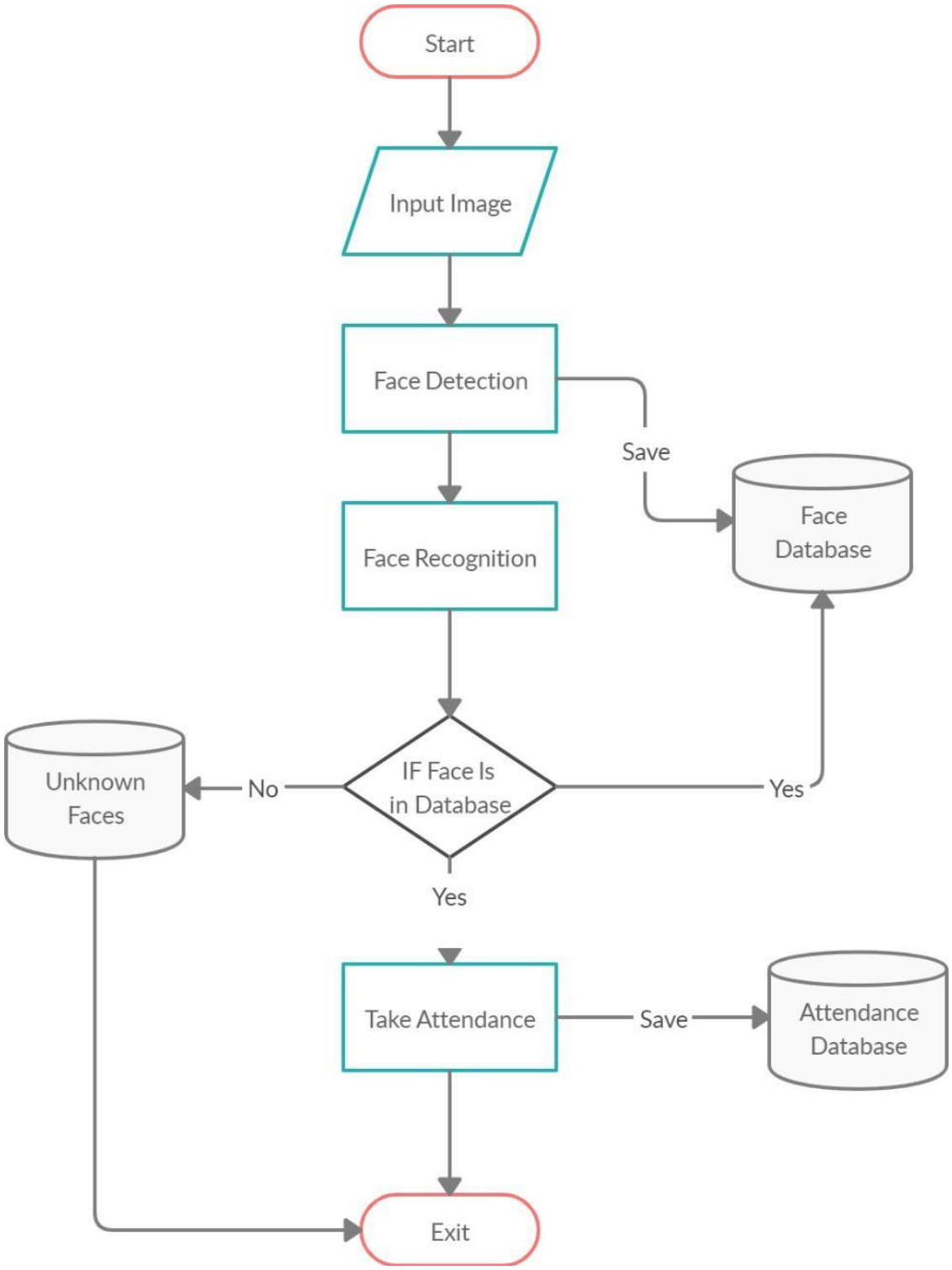
The objective of this project is to develop face recognition attendance system. Expected achievements in order to fulfill the objectives are:

- To detect the face segment from the video frame.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the identified student.

1.5. Scope of the project

We are setting up to design a system comprising of two modules. The first module (face detector) is a mobile component, which is basically a camera application that captures student faces and stores them in a file using computer vision face detection algorithms and face extraction techniques. The second module is a desktop application that does face recognition of the captured images (faces) in the file, marks the students register and then stores the results in a database for future analysis.

1.6. Flow Chart



Chapter 2

Literature Review

2.1. Student Attendance System

Disadvantages of RFID (Radio Frequency Identification) card system, fingerprint system and iris recognition system. RFID card system is implemented due to its simplicity. However, the user tends to help their friends to check in as long as they have their friend's ID card. The fingerprint system is indeed effective but not efficient because it takes time for the verification process so the user has to line up and perform the verification one by one. However for face recognition, the human face is always exposed and contain less information compared to iris. Iris recognition system which contains more detail might invade the privacy of the user. Voice recognition is available, but it is less accurate compared to other methods. Hence, face recognition system is suggested to be implemented in the student attendance system.

System Type	Advantage	Disadvantages
RFID card system	Simple	Fraudulent usage
Fingerprint system	Accurate	Time-consuming
Voice recognition system		Less accurate compared to Others
Iris recognition system	Accurate	Privacy Invasion

Table 2.1: Advantages & Disadvantages of Different Biometric System

2.2. Digital Image Processing

An image is defined as a two-dimensional function,**F(x,y)**, where x and y are spatial coordinates, and the amplitude of **F** at any pair of coordinates (x,y) is called the **intensity** of that image at that point. When x,y, and amplitude values of **F** are finite, we call it a **digital image**.Digital Image Processing is the processing of images which are digital in nature by a digital computer. Digital image processing techniques are motivated by three major applications mainly:

- Improvement of pictorial information for human perception
- Image processing for autonomous machine application
- Efficient storage and transmission.

Image processing mainly include the following steps:

1.Importing the image via image acquisition tools;

2.Analysing and manipulating the image;

3.Output in which result can be altered image or a report which is based on analysing that image

2.3. Image Representation in a Digital Computer

An image is a 2-Dimensional light intensity function

f (x,y) = r (x,y) × i (x,y) - (2.0)

Where,r (x, y) is the reflectivity of the surface of the corresponding image point. i (x,y) Represents the intensity of the incident light. A digital image f(x, y) is discretized both in spatial co-ordinates by grids and in brightness by quantization[3]. Effectively, the image can be represented as a matrix whose row, column indices specify a point in the image and the element value identifies gray level value at that point. These elements are referred to as pixels or pels.

Typically following image processing applications, the image size which is used is $\mathbf{256 \times 256}$, elements, $\mathbf{640 \times 480}$ pels or $\mathbf{1024 \times 1024}$ pixels. Quantization of these matrix pixels is done at 8 bits for black and white images and 24 bits for colored images (because of the three color planes Red, Green and Blue each at 8 bits).

Image as a Matrix

As we know, images are represented in rows and columns we have the following syntax in which images are represented as

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{bmatrix}$$

Figure: image matrix representation

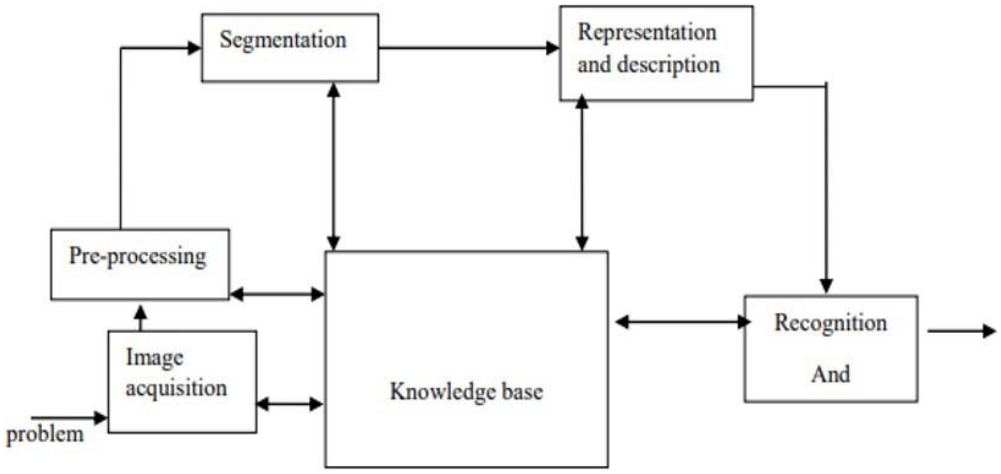
The right side of this equation is digital image by definition. Every element of this matrix is called image element , picture element , or pixel.

2.4. Steps in Digital Image Processing

Digital image processing involves the following basic tasks:

- Image Acquisition - An imaging sensor and the capability to digitize the signal produced by the sensor.
- Preprocessing – Enhances the image quality, filtering, contrast enhancement etc.
- Segmentation – Partitions an input image into constituent parts of objects.

- Description/feature Selection – extracts the description of image objects suitable for further computer processing.
- Recognition and Interpretation – Assigning a label to the object based on the information provided by its descriptor.
- Interpretation assigns meaning to a set of labelled objects.
- Knowledge Base – This helps for efficient processing as well as inter module cooperation.



2.5. **Definition of Terms and History**

Face Detection

Face detection is the process of identifying and locating all the present faces in a single image or video regardless of their position, scale, orientation, age and expression. Furthermore, the detection should be irrespective of extraneous illumination conditions and the image and video content.

Face Recognition

Face Recognition is a visual pattern recognition problem, where the face, represented as a three dimensional object that is subject to varying illumination, pose and other factors, needs to be identified based on acquired images.

Face Recognition is therefore simply the task of identifying an already detected face as a known or unknown face and in more advanced cases telling exactly whose face it is.

Difference between Face Detection and Face Recognition

Face detection answers the question, Where is the face? It identifies an object as a "face" and locates it in the input image. Face Recognition on the other hand answers the question who is this? Or whose face is it? It decides if the detected face is someone known or unknown based on the database of faces it uses to validate this input image[8].It can therefore be seen that face detections output (the detected face) is the input to the face recognizer and the face Recognition's output is the final decision i.e. face known or face unknown.

Face Detection

A face Detector has to tell whether an image of arbitrary size contains a human face and if so, where it is. Face detection can be performed based on several cues: skin color (for faces in color images and videos, motion (for faces in videos), facial/head shape, facial appearance or a combination of these parameters. Most face detection algorithms are appearance based without using other cues.

An input image is scanned at all possible locations and scales by a sub window. Face detection is posed as classifying the pattern in the sub window either as a face or a non-face. The face/nonface classifier is learned from face and non-face training examples using statistical learning methods.

Most modern algorithms are based on the Viola Jones object detection framework, which is based on Haar Cascades.

Face Detection Method	Advantages	Disadvantages
Viola Jones Algorithm	1. High detection Speed. 2. High Accuracy.	1. Long Training Time. 2.Limited Head Pose. 3.Not able to detect dark faces.
Local Binary Pattern Histogram	1.Simple computation. 2.High tolerance against the Monotonic 3.illumination changes.	1.Only used for binary and grey images. 2.Overall performance is inaccurate compared to Viola-Jones Algorithm.
Ada Boost Algorithm	Need not to have any prior knowledge about face structure.	The result highly depends on the training data and affected by weak classifiers.
SMQT Features and SNOW Classifier Method	1. Capable to deal with lighting problem in object detection. 2. Efficient in computation.	The region contain very similar to grey value regions will be misidentified as face.
Neural-Network	High accuracy only if large size of image were trained.	1. Detection process is slow and computation is complex. 2. Overall performance is weaker than Viola-Jones algorithm.

Viola-Jones algorithm which was introduced by P. Viola, M. J. Jones (2001) is the most popular algorithm to localize the face segment from static images or video frame. Basically the concept of Viola-Jones algorithm consists of four parts. The first part is known as Haar feature, second part is where integral image is created, followed by implementation of Adaboost on the third part and lastly cascading

process.

Viola-Jones algorithm analyses a given image using Haar features consisting of multiple rectangles .

In the fig shows several types of Haar features. The features perform as window function mapping onto the image. A single value result, which representing each feature can be computed by subtracting the sum of the white rectangle(s) from the sum of the black rectangle(s).

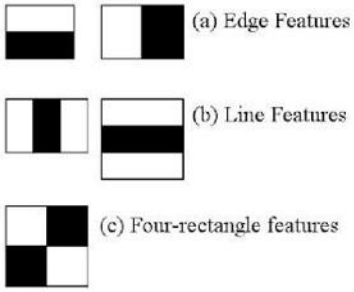


Figure : Haar Features

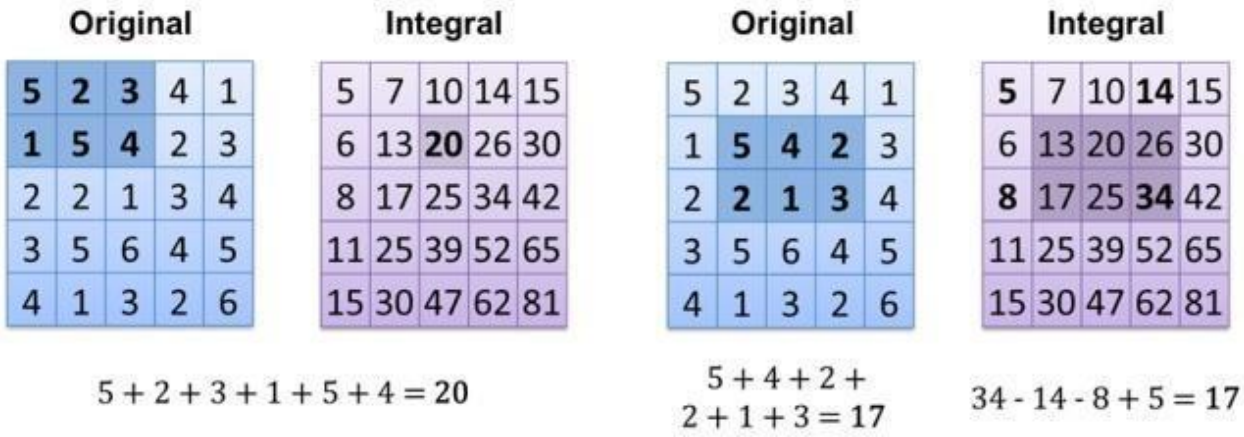


Figure: Integral of Image

The value of integrating image in a specific location is the sum of pixels on the left and the top of the respective location. In order to illustrate clearly, the value of the integral image at location 1 is the sum of the pixels in rectangle A. The values of integral image at the rest of the locations are cumulative.
For instance, the value at location 2 is summation of A and B, (A + B), at location 3 is summation of A and C, (A + C), and at location 4 is summation of all the regions, (A + B + C + D)

Local Binary Patterns Histogram

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture

classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets. Using the LBP combined with histograms we can represent the face images with a simple data vector.

LBPH algorithm work step by step:

LBPH algorithm work in 5 steps.

1. **Parameters:** the LBPH uses 4 parameters:
- **Radius:** the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
- **Neighbors:** the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
- **Grid X:** the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
- **Grid Y:** the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

2. **Training the Algorithm:** First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.
3. **Applying the LBP operation:** The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbors.
The image below shows this procedure:

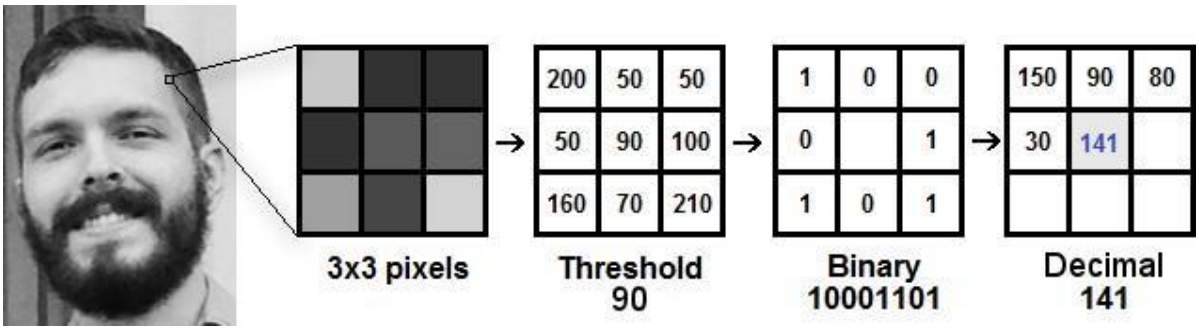


Figure: LBPHoperation

Based on the image above, let's break it into several small steps so we can understand it easily:

- Suppose we have a facial image in grayscale.
- We can get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).

- Then, we need to take the central value of the matrix to be used as the threshold.
- This value will be used to define the new values from the 8 neighbors.
- For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.
- Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concatenate the binary values (e.g. clockwise direction), but the final result will be the same.
- Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.

At the end of this procedure (LBP procedure), we have a new image which represents better the characteristics of the original image.

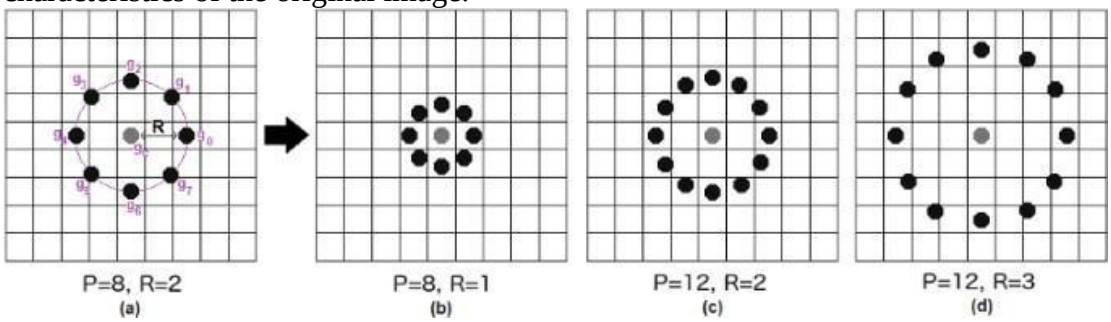


Figure: The LBP operation Radius Change

It can be done by using bilinear interpolation. If some data point is between the pixels, it uses the values from the 4 nearest pixels (2x2) to estimate the value of the new data point.

Extracting the Histograms: Now, using the image generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids, as can be seen in the following image:

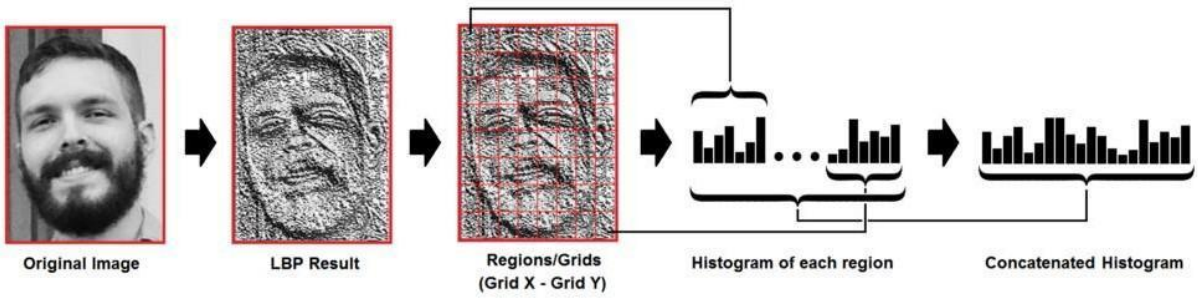


Figure : Extracting The Histogram

Based on the image above, we can extract the histogram of each region as follows:

- As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.
- Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

4. **Performing the face recognition:** In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.

- So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.
- We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: Euclidean distance, chi-square, absolute value, etc. In this example, we can use the **Euclidean distance** (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

Formula to calculate Distance between Histograms

- So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a 'confidence' measurement. Note: don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.
- We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

Chapter 3
Model Implementation & analysis

3.1. Introduction

Face detection involves separating image windows into two classes; one containing faces (turning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin color and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height).After taking the picture the system will compare the equality of the pictures in its database and give the most related result.

3.2. Model Implementation

The main components used in the implementation approach are open source computer vision library (OpenCV). One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. OpenCV library contains over 500 functions that span many areas in vision. The primary technology behind Face recognition is OpenCV. The user stands in front of the camera keeping a minimum distance of 50cm and his image is taken as an input. The frontal face is extracted from the image then converted to gray scale and stored. The Principal component Analysis (PCA) algorithm is performed on the images and the eigen values are stored in an xml file. When a user requests for recognition the frontal face is extracted from the captured video frame through the camera. The eigen value is re-calculated for the test face and it is matched with the stored data for the closest neighbor.

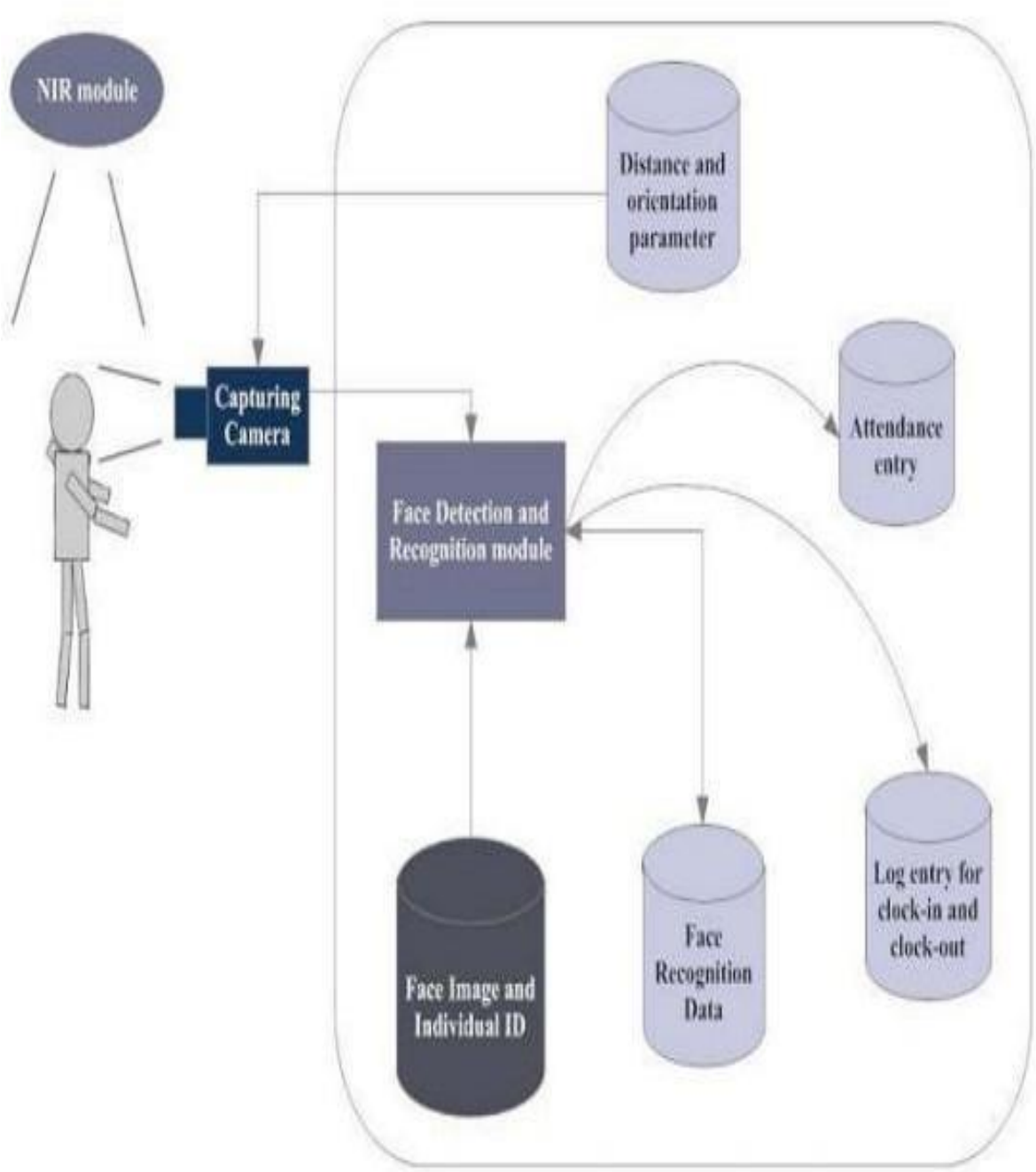


Figure : Model Implementation

3.3. Design Requirements

We used some tools to build the HFR system. Without the help of these tools it would not be possible to make it done. Here we will discuss about the most important one.

3.3.1. Software Implementation

1. **OpenCV:** We used OpenCV 3 dependency for python 3. OpenCV is library where there are lots of image processing functions are available. This is very useful library for image processing. Even one can get expected outcome without writing a single code. The library is cross-platform and free for use under the open-source BSD license. Example of some supported functions are given bellow:

- **Derivation**:Gradient/laplacian computing, contours delimitation
- **Hough transforms:**lines,segments,circles,and geometrical shapes detection
- **Histograms**: computing, equalization, and object localization with back projection algorithm
- **Segmentation**: thresholding, distance transform, foreground/ background detection, watershed segmentation
- **Filtering**: linear and nonlinear filters, morphological operations
- **Cascade detectors**: detection of face, eye, car plates

3.3.2. Interest points: detection and matching

- **Video processing:**optical flow, background subtraction, camshaft (object tracking)
- **Photography**:panoramas realization,high definition imaging (HDR), image inpaintingSo it was very important to install OpenCV. But installing OpenCV 3 is a complex process. How we did it is given below:

```
#!/bin/bash
#Usage : sudo bash./installopencv.bash
echo OpenCV 3.0.0 Raspbian Jessie auto install script - Thomas Cyrix
echo ================================================================
FILE="/tmp/out.$$"
GREP="/bin/grep"
if [ "$(id -u)" != "0" ]; then
   echo "This script must be run as root" 1>&2
   exit 1
fi
echo installing core dependencies ...
apt-get -y install cmake python3-dev python3.4-dev python3-numpy gcc build-essential cmake-curses-gui
echo installing other dependencies ...
apt-get -y install pkg-config libpng12-0 libpng12-dev libpng++-dev libpng3 libpnglite-dev zlib1g-dbg zlib1g
zlib1g-dev pngtools libtiff5-dev libtiff5 libtiffxx0c2 libtiff-tools libeigen3-dev
echo installing helper apps ...
apt-get -y libav-tools
#apt-get -y ffmpeg libavcodec55 libavformat55
apt-get -y install libjpeg8 libjpeg8-dbg libjpeg8-dbg libjpeg-progs libavcodec-dev libavformat-dev libgstreamer0.10-0-dbg
libgstreamer0.10-0 libgstreamer0.10-dev libxine2-ffmpeg libxine2-dev libxine2-bin libunicap2 libunicap2-dev swig libv4l-0 libv4l-dev libpython3.4 libgtk2.0-dev
echo Receving OpenCV 3.0.0 source...
git clone --branch 3.0.0 --depth 1 https://github.com/Itseez/opencv.git
cd opencv
mkdir release
cd release
echo Preparing compilation, may take a long while...
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=$(python3 -c "import sys; print(sys.prefix)") -D PYTHON_EXECUTABLE=$(which python3) ..
echo Compiling Open CV 3.0.0, may take 2 to 36 hours
make -j4
echo Compilation Ok, installing...
make install
cd ../..
rm -r opencv
echo Completed !
echo You now can use OpenCV 3.0.0 in both Python 2 and Python 3 !
```

We copied this script and place it on a directory on our raspberry pi and saved it. Then through terminal we made this script executable and then ran it.

```
pip install opencv-python
```

these are the command line we used.

2. **Python IDE:** There are lots of IDEs for python. Some of them are PyCharm, Thonny, Ninja, Spyder etc.

3. PyCharm deeply understands your project, not just individual files

Refactoring is a breeze across an entire project

The built-in SQL tooling is amazing

Autocomplete works better than any other editor, by far

We installed PyCharm IDE.

```
1. sudo apt-get isntall spyder
```

3.4. Hardware Interfaces

Hardware Requirement	Core 2 Duo 2nd generation
System Configuration	RAM – 2 GB DDR – 3
Operating System	Windows – 7/8/8.1/10
Other hardwares	D3D D8801 Home Security Camera (128 GB, 1 Channel)

Software Interfaces

Front End	Python
Back End	Python ,Open CV

3.5. Experimental Results

The step of the experiments process are given below:

Face Detection:

Start capturing images through web camera of the client side: Begin:

- Pre-process the captured image and extract face image
- calculate the eigen value of the captured face image and compared with eigen values of existing faces in the database.
- If eigen value does not matched with existing ones,save the new face image information to the face database (xml file).
- If eigen value matched with existing one then recognition step will done.

End

Face Recognition:

Local Binary Patterns Histogram (LBPH)

Local Binary Patterns Histogram algorithm was proposed in 2006. It is based on local binary operator. It is widely used in facial recognition due to its computational simplicity and discriminative power.

The steps involved to achieve this are:

- creating dataset
- face acquisition
- feature extraction
- classification

The LBPH algorithm is a part of opencv.

Steps

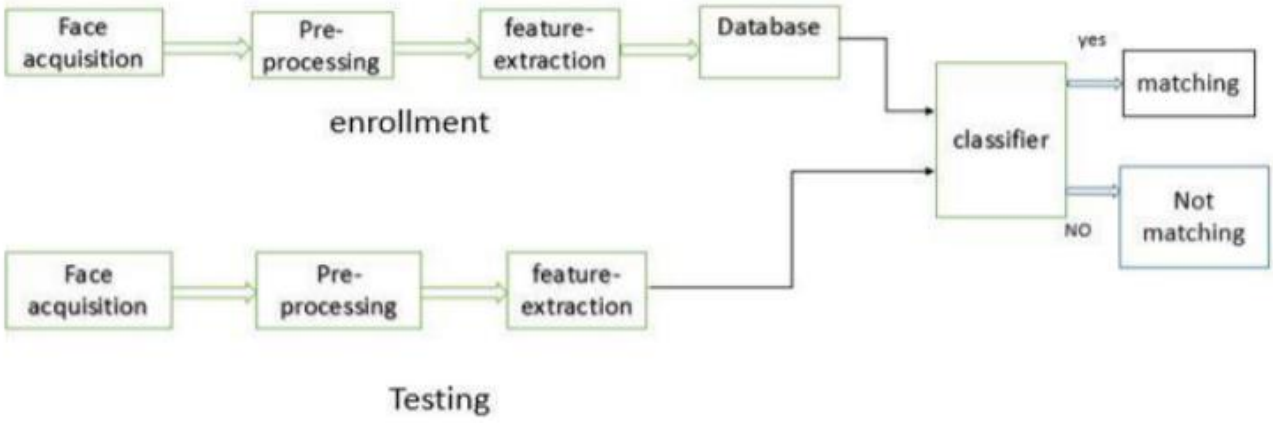


Figure: LPBH Algorithm

- Suppose we have an image having dimentions N x M.
- We divide it into regions of same height and width resulting in m x m dimension for every region.

Figure:Face Segmentation

- Local binary operator is used for every region. The LBP operator is defined in window of 3x3.

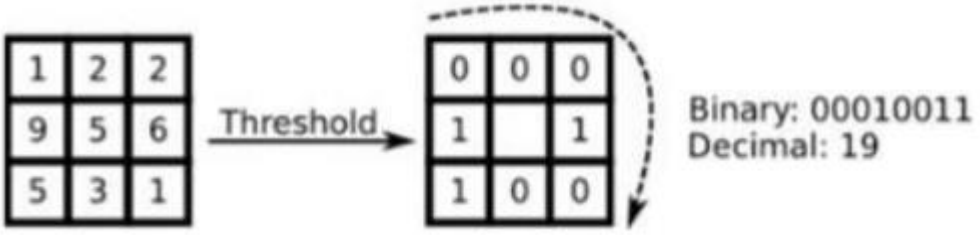
$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

here '(Xc,Yc)' is central pixel with intensity 'Ic'. And 'In' being the intensity of the the neighbor pixel

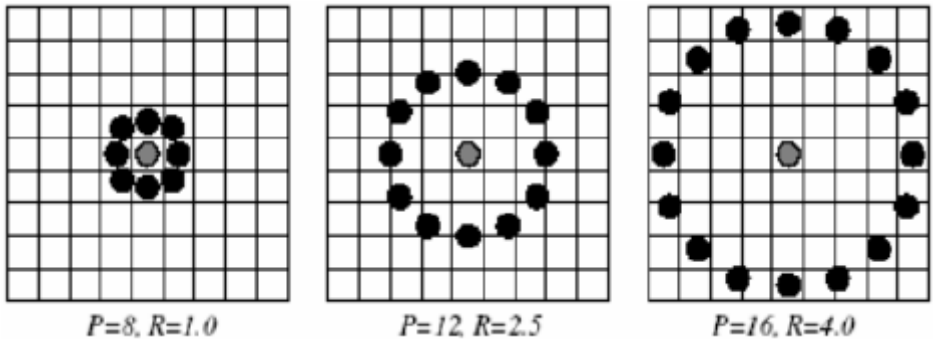
- Using median pixel value as threshold, it compares a pixel to its 8 closest pixels using this function.

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

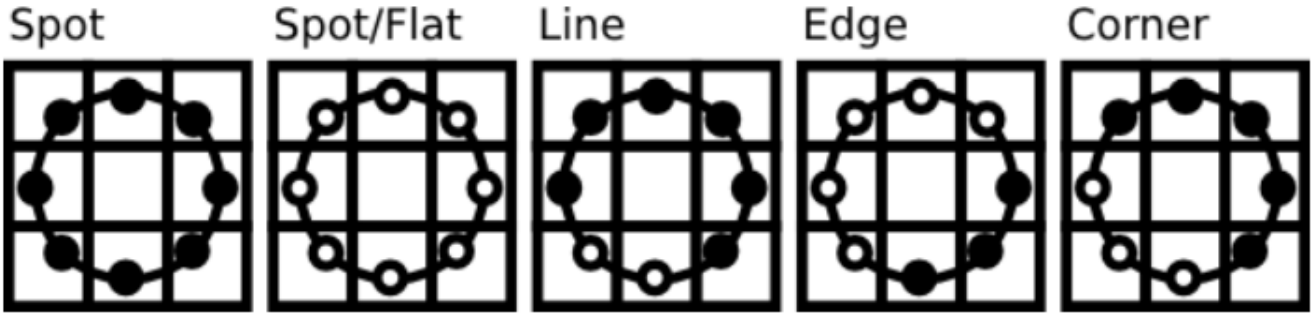
- If the value of neighbor is greater than or equal to the central value it is set as 1 otherwise it is set as 0.
- Thus, we obtain a total of 8 binary values from the 8 neighbors.
- After combining these values we get a 8 bit binary number which is translated to decimal number for our convenience.
- This decimal number is called the pixel LBP value and its range is 0-255.



- Later it was noted that a fixed neighborhood fails to encode details varying in scale .The algorithm was improved to use different number of radius and neighbors , now it was known as circular LBP.



- The idea here is to align an arbitrary number of neighbors on a circle with a variable radius. This way the following neighborhoods are captured:



- For a given point (Xc,Yc) the position of the neighbor (Xp,Yp), p belonging to P can be calculated by:

$$x_p = x_c + R\cos(\frac{2\pi p}{P})$$

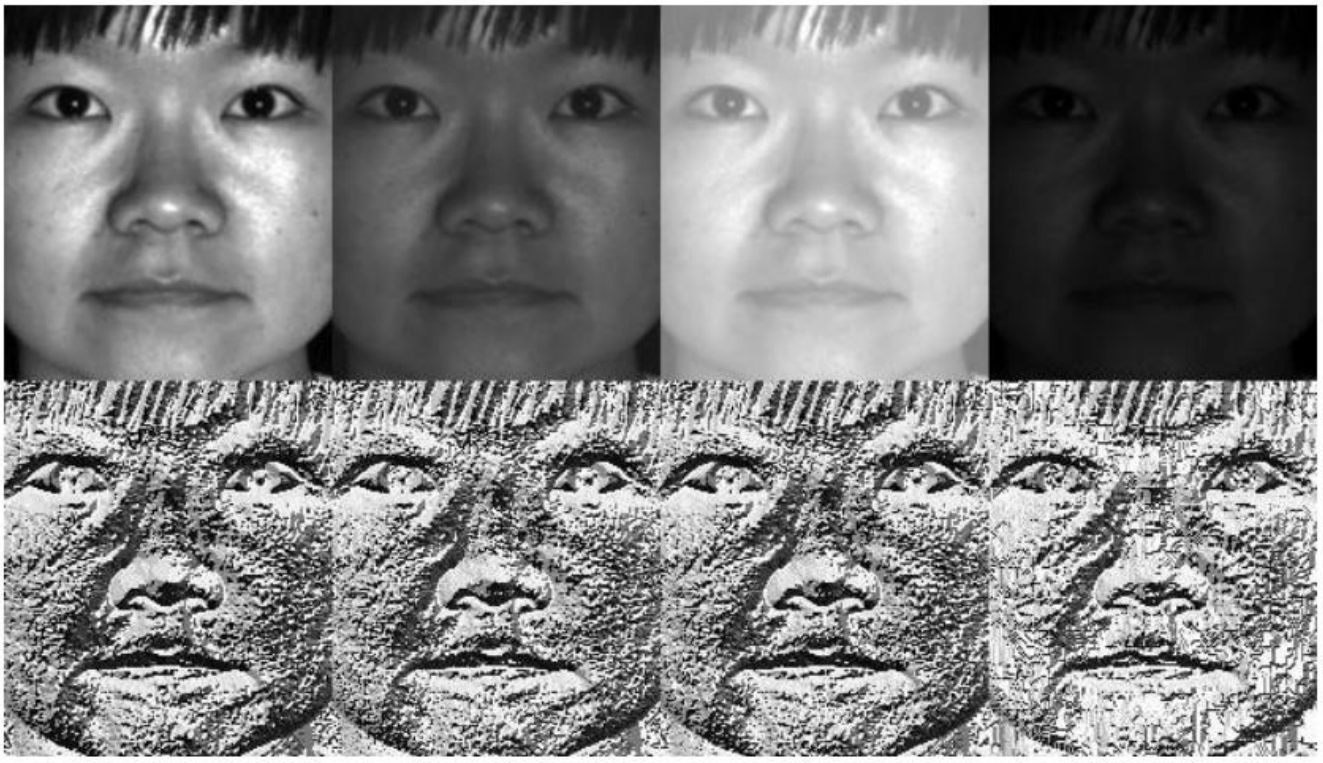
$$y_p = y_c - R\sin(\frac{2\pi p}{P})$$

here R is radius of the circle and P is the number of sample points.

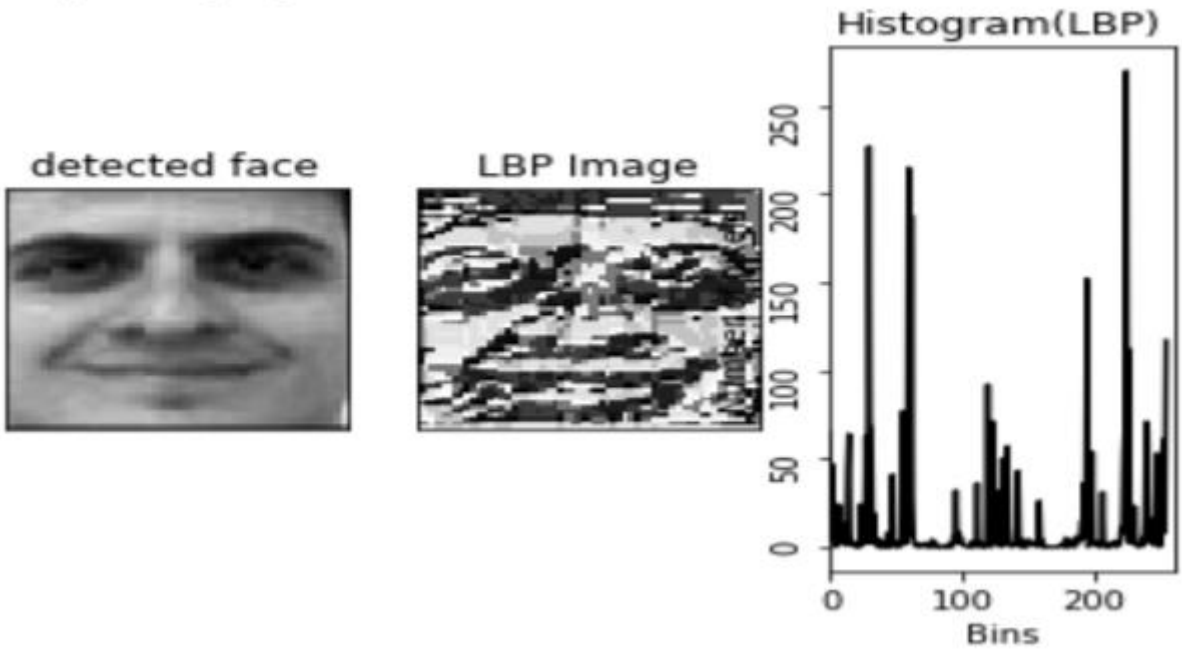
- If a points coordinate on the circle doesn't correspond to image coordinates, it get's interpolated generally by bilinear interpolation:

$$f(x,y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}$$

- The LBP operator is robust against monotonic gray scale transformations.



- After the generation of LBP value histogram of the region is created by counting the number of similar LBP values in the region.
- After creation of histogram for each region all the histograms are merged to form a single histogram and this is known as feature vector of the image.



15lpb histogram

- Now we compare the histograms of the test image and the images in the database and then we return the image with the closest hitogram.
(This can be done using many techniques like euclidean distance, chi-square, absolute value etc)
- The Euclidean distance is calculated by comparing the test image features with features stored in the dataset.The minimum distance between test and original image gives the matching rate.

$$d(a,b) = \sqrt{\sum_{i=1}^{n} |a_i - b_i|^2}$$

- As an output we get an ID of the image from the database if the test image is recognised.

LBPH can recognise both side and front faces and it is not affected by illumination variations which means that it is more flexible.

Implementation

- The dataset can be created by taking images from webcam or from saved images. We will take many samples of a single person. An unique ID or a name is given to a person in the database.

Import the necessary libraries

import numpy as np

import cv2

import os

Python

- Creating database by adding names of peopple we wanna add to database

creating database

database = ["Tom Cruise", "Clinton"]

- Face detection using Haarcascade Classifier, cropping the face and grayscaling the face .

```
def face_detection(image):
    image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    haar_classifier = cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_default.xml')
    face = haar_classifier.detectMultiScale(image_gray, scaleFactor=1.3, minNeighbors=7)
    (x,y,w,h) = face[0]
    return image_gray[y:y+w, x:x+h], face[0]
```

Python

- Preparing data with labels and faces

```
def prepare_data(data_path):
    folders = os.listdir(data_path)
    labels = []
    faces = []
    for folder in folders:
        label = int(folder)
        training_images_path = data_path + '/' + folder
        for image in os.listdir(training_images_path):
            image_path = training_images_path + '/' + image
            training_image = cv2.imread(image_path)
            face, bounding_box = face_detection(training_image)
            faces.append(face)
            labels.append(label)

    print ('Training Done')
```

```
    return faces, labels
```

Python

```
faces, labels = prepare_data('training')
print ('Total faces = ', len(faces))
print ('Total labels = ', len(labels))
```

Python

- Creating LBPH model and training it with the prepared data

```
model = cv2.face.createLBPHFaceRecognizer()
model.train(faces, np.array(labels))
```

- Testing the trained model using a test image

```
def predict_image(test_image):
    img = test_image.copy()
    face, bounding_box = face_detection(img)
    label = model.predict(face)
    label_text = database[label-1]
    print (label)
    print (label_text)
    (x,y,w,h) = bounding_box
    cv2.rectangle(img, (x,y), (x+w, y+h), (0,255,0), 2)
    cv2.putText(img, label_text, (x,y), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 255, 0), 2)
    return img
  test1 = cv2.imread("test/tom.jpg")
predict1 = predict_image(test1)
cv2.imshow('Face Recognition', predict1)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Advantages of LBPH algorithm

- LBPH Method is one of the best performing texture descriptor.
- The LBP operator is robust against monotonic gray scale transformations.
- FisherFaces only prevents features of a person from becoming dominant, but it still considers illumination variations as a useful feature. But light variation is not a useful feature to extract as it is not part of actual face
- Fisherfaces need larger storage of face data and more processing time in recognition.
- In LBPH each image is analyzed independently, while the eigenfaces and fisherfaces method

looks at the dataset as a whole.

- LBPH method will probably work better than fisherfaces in different environments and light conditions .It also depends on our training and testing data sets.
- It can represent local features in the images.
- LBPH can recognise both side and front faces.

Conclusion

- Computer vision is a major part of every big project relates to security, banking, marketing etc.
- LBHP algorithm can be use in various applications for facial recognition.
- It is very easy to implement.

Here is our data set sample.

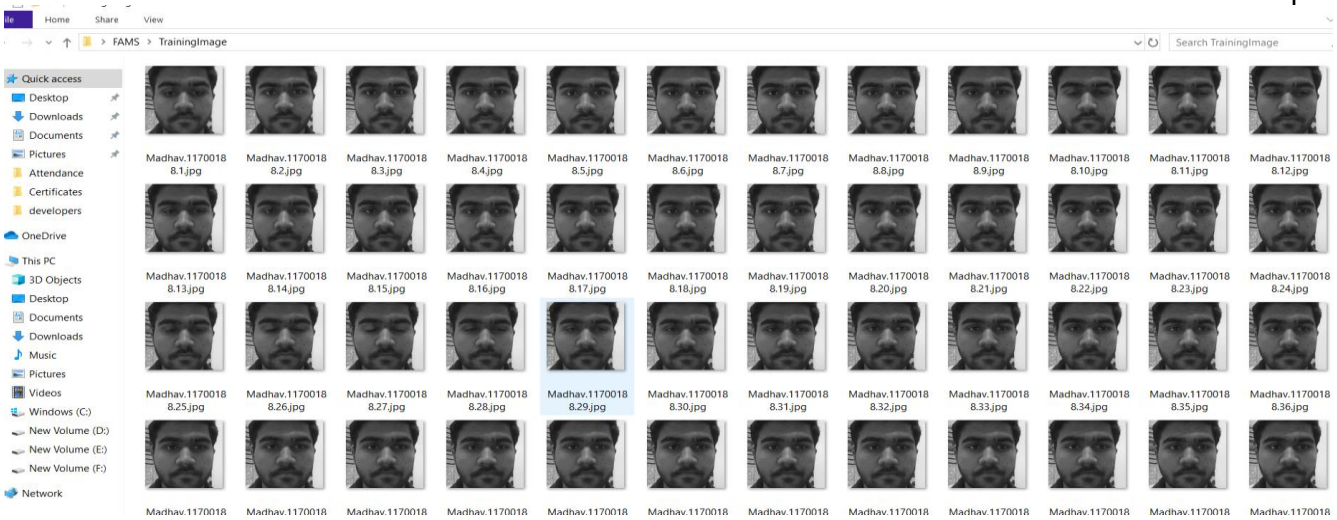


Figure: Sample Dataset

Face Orientations	Detection Rate	Recognition Rate
O º (Frontal face)	98.7 %	95%
18º	80.0 %	78%
54º	59.2 %	58%
72º	0.00 %	0.00%
90º(Profile face)	0.00 %	0.00%

We performed a set of experiments to demonstrate the efficiency of the proposed method. 30 different images of 10 persons are used in training set. Figure 3 shows a sample binary image detected by the ExtractFace() function using Paul-Viola Face extracting Frame work detection method.

Chapter 4

Code Implementation

4.1. Code Implementation

All our code is written in Python language. First here is our project directory structure and files.

```
 Attendance
Datasets
Developers
StudenDetails
TrainingImage
TrainingImageLabel
AMS.png
AMS_run.py
GUI_RS.py
Haarcascade_frontalface_alt.xml
haarcascade_frontalface_default.xml
Mail.py
Register_student.py
retrain.py
test.py
test2.py
test3.py
testing.py
training.py:
```

```
Dataset: Where all the faces are saved.
main.py: Main program file to run the program.
dataset.py: Capture images and working on datasets.
database.log: To keep track the database events
data_set.csv: To save the details of data.
data_log.ods: Attendance save
```

```
import tkinter as tk
from tkinter import *
import webbrowser
import cv2
import csv
import os
import numpy as np
from PIL import Image,ImageTk
import pandas as pd
import datetime
import time
```

```
#####Window is our Main frame of system
window = tk.Tk()
window.title("FAMS-Face Recognition Based Attendance Management System")

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

####GUI for manually fill attendance

def manually_fill():
    global sb
    sb = tk.Tk()
    sb.iconbitmap(r"C:/Users/Madhav Sharma/Desktop/FAMS/AMS.ico")
    sb.title("Enter subject name...")
    sb.geometry('580x320')
    sb.configure(background='snow')

    def err_screen_for_subject():

        def ec_delete():
            ec.destroy()
        global ec
        ec = tk.Tk()
        ec.geometry('300x100')
        ec.iconbitmap(r"C:/Users/Madhav Sharma/Desktop/FAMS/AMS.ico")
        ec.title('Warning!!')
        ec.configure(background='snow')
        Label(ec, text='Please enter your subject name!!!', fg='red', bg='white', font=('times', 16, ' bold
')).pack()
        Button(ec, text='OK', command=ec_delete, fg="black", bg="lawn green", width=9, height=1,
activebackground="Red",
               font=('times', 15, ' bold ')).place(x=90, y=50)

    def fill_attendance():
        ts = time.time()
        Date = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
        timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
        Time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
        Hour, Minute, Second = timeStamp.split(":")
        ####Creatting csv of attendance

        ##Create table for Attendance
        date_for_DB = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
        global subb
        subb=SUB_ENTRY.get()
```

```
DB_table_name = str(subb + "_" + Date + "_Time_" + Hour + "_" + Minute + "_" + Second)

import pymysql.connections

###Connect to the database
try:
    global cursor
    connection = pymysql.connect(host='localhost', user='root', password='madhav', db='test')
    cursor = connection.cursor()
except Exception as e:
    print(e)

sql = "CREATE TABLE " + DB_table_name + """
              (ID INT NOT NULL AUTO_INCREMENT,
               ENROLLMENT varchar(100) NOT NULL,
               NAME VARCHAR(50) NOT NULL,
               DATE VARCHAR(20) NOT NULL,
               TIME VARCHAR(20) NOT NULL,
                  PRIMARY KEY (ID)
                  );
             """


try:
    cursor.execute(sql)  ##for create a table
except Exception as ex:
    print(ex)  #

if subb==":
    err_screen_for_subject()
else:
    sb.destroy()
    MFW = tk.Tk()
    MFW.iconbitmap(r"C:/Users/Madhav Sharma/Desktop/FAMS/AMS.ico")
    MFW.title("Manually attendance of "+ str(subb))
    MFW.geometry('880x470')
    MFW.configure(background='snow')

    def del_errsc2():
        errsc2.destroy()

    def err_screen1():
        global errsc2
        errsc2 = tk.Tk()
        errsc2.geometry('330x100')
        errsc2.iconbitmap(r'C:/Users/Madhav Sharma/Desktop/FAMS/AMS.ico')
```

```
        errsc2.title('Warning!!')
        errsc2.configure(background='snow')
        Label(errsc2, text='Please enter Student & Enrollment!!!', fg='red', bg='white',
              font=('times', 16, ' bold ')).pack()
        Button(errsc2, text='OK', command=del_errsc2, fg="black", bg="lawn green", width=9,
height=1,
               activebackground="Red", font=('times', 15, ' bold ')).place(x=90, y=50)

    def testVal(inStr, acttyp):
        if acttyp == '1':  # insert
            if not inStr.isdigit():
                return False
        return True

    ENR = tk.Label(MFW, text="Enter Enrollment", width=15, height=2, fg="white", bg="blue2",
                   font=('times', 15, ' bold '))
    ENR.place(x=30, y=100)

    STU_NAME = tk.Label(MFW, text="Enter Student name", width=15, height=2, fg="white",
bg="blue2",
                        font=('times', 15, ' bold '))
    STU_NAME.place(x=30, y=200)

    global ENR_ENTRY
    ENR_ENTRY = tk.Entry(MFW, width=20,validate='key', bg="yellow", fg="red", font=('times',
23, ' bold '))
    ENR_ENTRY['validatecommand'] = (ENR_ENTRY.register(testVal), '%P', '%d')
    ENR_ENTRY.place(x=290, y=105)

    def remove_enr():
        ENR_ENTRY.delete(first=0, last=22)

    STUDENT_ENTRY = tk.Entry(MFW, width=20, bg="yellow", fg="red", font=('times', 23, ' bold
'))
    STUDENT_ENTRY.place(x=290, y=205)

    def remove_student():
        STUDENT_ENTRY.delete(first=0, last=22)

    ####get important variable
    def enter_data_DB():
        ENROLLMENT = ENR_ENTRY.get()
        STUDENT = STUDENT_ENTRY.get()
        if ENROLLMENT==":
            err_screen1()
        elif STUDENT==":
```

```
            err_screen1()
        else:
            time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            Hour, Minute, Second = time.split(":")
            Insert_data = "INSERT INTO " + DB_table_name + "
(ID,ENROLLMENT,NAME,DATE,TIME) VALUES (0, %s, %s, %s,%s)"
            VALUES = (str(ENROLLMENT), str(STUDENT), str(Date), str(time))
            try:
                cursor.execute(Insert_data, VALUES)
            except Exception as e:
                print(e)
            ENR_ENTRY.delete(first=0, last=22)
            STUDENT_ENTRY.delete(first=0, last=22)

    def create_csv():
        import csv
        cursor.execute("select * from " + DB_table_name + ";")
        csv_name=r'C:/Users/Madhav Sharma/Desktop/FAMS/Attendance/Manually
Attendance/'+DB_table_name+'.csv'
        with open(csv_name, "w") as csv_file:
            csv_writer = csv.writer(csv_file)
            csv_writer.writerow([i[0] for i in cursor.description])  # write headers
            csv_writer.writerows(cursor)
            O="CSV created Successfully"
            Notifi.configure(text=O, bg="Green", fg="white", width=33, font=('times', 19, 'bold'))
            Notifi.place(x=180, y=380)
        import csv
        import tkinter
        root = tkinter.Tk()
        root.title("Attendance of " + subb)
        root.configure(background='snow')
        with open(csv_name, newline="") as file:
            reader = csv.reader(file)
            r = 0

            for col in reader:
                c = 0
                for row in col:
                    # i've added some styling
                    label = tkinter.Label(root, width=13, height=1, fg="black", font=('times', 13, ' bold '),
                                          bg="lawn green", text=row, relief=tkinter.RIDGE)
                    label.grid(row=r, column=c)
                    c += 1
                r += 1
        root.mainloop()
```

```
        Notifi = tk.Label(MFW, text="CSV created Successfully", bg="Green", fg="white", width=33,
                          height=2, font=('times', 19, 'bold'))


        c1ear_enroll = tk.Button(MFW, text="Clear", command=remove_enr, fg="black", bg="deep
pink", width=10,
                                 height=1,
                                 activebackground="Red", font=('times', 15, ' bold '))
        c1ear_enroll.place(x=690, y=100)

        c1ear_student = tk.Button(MFW, text="Clear", command=remove_student, fg="black",
bg="deep pink", width=10,
                                  height=1,
                                  activebackground="Red", font=('times', 15, ' bold '))
        c1ear_student.place(x=690, y=200)

        DATA_SUB = tk.Button(MFW, text="Enter Data",command=enter_data_DB, fg="black",
bg="lime green", width=20,
                             height=2,
                             activebackground="Red", font=('times', 15, ' bold '))
        DATA_SUB.place(x=170, y=300)

        MAKE_CSV = tk.Button(MFW, text="Convert to CSV",command=create_csv, fg="black",
bg="red", width=20,
                             height=2,
                             activebackground="Red", font=('times', 15, ' bold '))
        MAKE_CSV.place(x=570, y=300)

        def attf():
            import subprocess
            subprocess.Popen(r'explorer                                    /select,"C:/Users/Madhav
Sharma/Desktop/FAMS/Attendance/Manually Attendance/-------Check atttendance-------"')

        attf = tk.Button(MFW,  text="Check Sheets",command=attf,fg="black"   ,bg="lawn green"
,width=12  ,height=1 ,activebackground = "Red" ,font=('times', 14, ' bold '))
        attf.place(x=730, y=410)

        MFW.mainloop()


    SUB = tk.Label(sb, text="Enter Subject", width=15, height=2, fg="white", bg="blue2", font=('times',
15, ' bold '))
    SUB.place(x=30, y=100)

    global SUB_ENTRY
```

```
    SUB_ENTRY = tk.Entry(sb, width=20, bg="yellow", fg="red", font=('times', 23, ' bold '))
    SUB_ENTRY.place(x=250, y=105)

    fill_manual_attendance = tk.Button(sb, text="Fill Attendance",command=fill_attendance, fg="white",
bg="deep pink", width=20, height=2,
                       activebackground="Red", font=('times', 15, ' bold '))
    fill_manual_attendance.place(x=250, y=160)
    sb.mainloop()


##For clear textbox
def clear():
    txt.delete(first=0, last=22)

def clear1():
    txt2.delete(first=0, last=22)
def del_sc1():
    sc1.destroy()
def err_screen():
    global sc1
    sc1 = tk.Tk()
    sc1.geometry('300x100')
    sc1.iconbitmap('C:/Users/Madhav Sharma/Desktop/FAMS/AMS.ico')
    sc1.title('Warning!!')
    sc1.configure(background='snow')
    Label(sc1,text='Enrollment & Name required!!!',fg='red',bg='white',font=('times', 16, ' bold ')).pack()
    Button(sc1,text='OK',command=del_sc1,fg="black"      ,bg="lawn green"      ,width=9      ,height=1,
activebackground = "Red" ,font=('times', 15, ' bold ')).place(x=90,y= 50)

##Error screen2
def del_sc2():
    sc2.destroy()
def err_screen1():
    global sc2
    sc2 = tk.Tk()
    sc2.geometry('300x100')
    sc2.iconbitmap('C:/Users/Madhav Sharma/Desktop/FAMS/AMS.ico')
    sc2.title('Warning!!')
    sc2.configure(background='snow')
    Label(sc2,text='Please enter your subject name!!!',fg='red',bg='white',font=('times', 16, ' bold ')).pack()
    Button(sc2,text='OK',command=del_sc2,fg="black"      ,bg="lawn green"      ,width=9      ,height=1,
activebackground = "Red" ,font=('times', 15, ' bold ')).place(x=90,y= 50)

###For take images for datasets
def take_img():
    l1 = txt.get()
```

```
l2 = txt2.get()
if l1 == '':
    err_screen()
elif l2 == '':
    err_screen()
else:
    try:
        cam = cv2.VideoCapture(0)
        detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        Enrollment = txt.get()
        Name = txt2.get()
        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
                cv2.imwrite("TrainingImage/ " + Name + "." + Enrollment + '.' + str(sampleNum) + ".jpg",
                            gray[y:y + h, x:x + w])
                cv2.imshow('Frame', img)
            # wait for 100 miliseconds
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
            # break if the sample number is morethan 100
            elif sampleNum > 200:
                break
        cam.release()
        cv2.destroyAllWindows()
        ts = time.time()
        Date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
        Time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
        row = [Enrollment, Name, Date, Time]
        with open('StudentDetails/StudentDetails.csv', 'a+') as csvFile:
            writer = csv.writer(csvFile, delimiter=',')
            writer.writerow(row)
            csvFile.close()
        res = "Images Saved for Enrollment : " + Enrollment + " Name : " + Name
        Notification.configure(text=res, bg="SpringGreen3", width=50, font=('times', 18, 'bold'))
        Notification.place(x=250, y=400)
    except FileExistsError as F:
        f = 'Student Data already exists'
```

```
        Notification.configure(text=f, bg="Red", width=21)
        Notification.place(x=450, y=400)


###for choose subject and fill attendance
def subjectchoose():
    def Fillattendances():
        sub=tx.get()
        now = time.time()  ###For calculate seconds of video
        future = now + 20
        if time.time() < future:
            if sub == '':
                err_screen1()
            else:
                recognizer = cv2.face.LBPHFaceRecognizer_create()  # cv2.createLBPHFaceRecognizer()
                try:
                    recognizer.read("TrainingImageLabel/Trainner.yml")
                except:
                    e = 'Model not found,Please train model'
                    Notifica.configure(text=e, bg="red", fg="black", width=33, font=('times', 15, 'bold'))
                    Notifica.place(x=20, y=250)

                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 = ['Enrollment', '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:
                        global Id

                        Id, conf = recognizer.predict(gray[y:y + h, x:x + w])
                        if (conf > 20):
                            print(conf)
                            global Subject
                            global aa
                            global date
                            global timeStamp
                            Subject = tx.get()
                            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['Enrollment'] == Id]['Name'].values
                global tt
                tt = str(Id) + "-" + aa
                En = '15624031' + str(Id)
                attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]
                cv2.rectangle(im, (x, y), (x + w, y + h), (0, 260, 0), 7)
                cv2.putText(im, str(tt), (x + h, y), font, 1, (255, 255, 0,), 4)

            else:
                Id = 'Unknown'
                tt = str(Id)
                cv2.rectangle(im, (x, y), (x + w, y + h), (0, 25, 255), 7)
                cv2.putText(im, str(tt), (x + h, y), font, 1, (0, 25, 255), 4)
        if time.time() > future:
            break

        attendance = attendance.drop_duplicates(['Enrollment'], keep='first')
        cv2.imshow('Filling attedance..', im)
        key = cv2.waitKey(30) & 0xff
        if key == 27:
            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(":")
    ####Creatting csv of attendance
    fileName = "Attendance/" + Subject + "_" + date + "_" + Hour + "-" + Minute + "-" + Second
+ ".csv"
    attendance.to_csv(fileName, index=False)

    ##Create table for Attendance
    date_for_DB = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
    DB_Table_name = str( Subject + "_" + date_for_DB + "_Time_" + Hour + "_" + Minute + "_"
+ Second)
    import pymysql.connections

    ###Connect to the database
    try:
        global cursor
        connection = pymysql.connect(host='localhost', user='root', password='', db='test')
        cursor = connection.cursor()
    except Exception as e:
        print(e)
```

```
            sql = "CREATE TABLE " + DB_Table_name + """
            (ID INT NOT NULL AUTO_INCREMENT,
             ENROLLMENT varchar(100) NOT NULL,
             NAME VARCHAR(50) NOT NULL,
             DATE VARCHAR(20) NOT NULL,
             TIME VARCHAR(20) NOT NULL,
                PRIMARY KEY (ID)
                );
            """
            ####Now enter attendance in Database
            insert_data = "INSERT INTO " + DB_Table_name + "
(ID,ENROLLMENT,NAME,DATE,TIME) VALUES (0, %s, %s, %s,%s)"
            VALUES = (str(Id), str(aa), str(date), str(timeStamp))
            try:
               cursor.execute(sql)  ##for create a table
               cursor.execute(insert_data, VALUES)##For insert data into table
            except Exception as ex:
               print(ex)  #

            M = 'Attendance filled Successfully'
            Notifica.configure(text=M, bg="Green", fg="white", width=33, font=('times', 15, 'bold'))
            Notifica.place(x=20, y=250)

            cam.release()
            cv2.destroyAllWindows()

            import csv
            import tkinter
            root = tkinter.Tk()
            root.title("Attendance of " + Subject)
            root.configure(background='snow')
            cs = 'C:/Users/Madhav Sharma/Desktop/FAMS' + fileName
            global attach
            attach=cs
            with open(cs, newline="") as file:
               reader = csv.reader(file)
               r = 0

               for col in reader:
                  c = 0
                  for row in col:
                     # i've added some styling
                     label = tkinter.Label(root, width=8, height=1, fg="black", font=('times', 15, ' bold '),
                                           bg="lawn green", text=row, relief=tkinter.RIDGE)
                     label.grid(row=r, column=c)
                     c += 1
```

```
                r += 1
            root.mainloop()
            print(attendance)

    ###windo is frame for subject chooser
    windo = tk.Tk()
    windo.iconbitmap('C:/Users/Madhav Sharma/Desktop/FAMS/AMS.ico')
    windo.title("Enter subject name...")
    windo.geometry('580x320')
    windo.configure(background='snow')

    Notifica = tk.Label(windo, text="Attendance filled Successfully", bg="Green", fg="white", width=33,
                        height=2, font=('times', 15, 'bold'))

    def Attf():
        import subprocess
        subprocess.Popen(r'explorer  /select,"C:/Users/Madhav  Sharma/Desktop/FAMS/Attendance/-------
Check atttendance-------"')

    attf = tk.Button(windo,  text="Check Sheets",command=Attf,fg="black"  ,bg="lawn green"  ,width=12
,height=1 ,activebackground = "Red" ,font=('times', 14, ' bold '))
    attf.place(x=430, y=255)


    sub = tk.Label(windo, text="Enter Subject", width=15, height=2, fg="white", bg="blue2",
font=('times', 15, ' bold '))
    sub.place(x=30, y=100)

    tx = tk.Entry(windo, width=20, bg="yellow", fg="red", font=('times', 23, ' bold '))
    tx.place(x=250, y=105)

    fill_a = tk.Button(windo, text="Fill Attendance", fg="white",command=Fillattendances, bg="deep
pink", width=20, height=2,
                       activebackground="Red", font=('times', 15, ' bold '))
    fill_a.place(x=250, y=160)
    windo.mainloop()

def admin_panel():
    win = tk.Tk()
    win.iconbitmap('C:/Users/Madhav Sharma/Desktop/FAMS/AMS.ico')
    win.title("LogIn")
    win.geometry('880x420')
    win.configure(background='snow')

    def log_in():
        username = un_entr.get()
```

```
    password = pw_entr.get()

    if username == 'admin':
        if password == 'admin':
            win.destroy()
            import csv
            import tkinter
            root = tkinter.Tk()
            root.title("Student Details")
            root.configure(background='snow')

            cs = 'C:/Users/Madhav Sharma/Desktop/FAMS/StudentDetails/StudentDetails.csv'
            with open(cs, newline="") as file:
                reader = csv.reader(file)
                r = 0

                for col in reader:
                    c = 0
                    for row in col:
                        # i've added some styling
                        label = tkinter.Label(root, width=8, height=1, fg="black", font=('times', 15, ' bold '),
                                              bg="lawn green", text=row, relief=tkinter.RIDGE)
                        label.grid(row=r, column=c)
                        c += 1
                    r += 1
            root.mainloop()
        else:
            valid = 'Incorrect ID or Password'
            Nt.configure(text=valid, bg="red", fg="black", width=38, font=('times', 19, 'bold'))
            Nt.place(x=120, y=350)

    else:
        valid ='Incorrect ID or Password'
        Nt.configure(text=valid, bg="red", fg="black", width=38, font=('times', 19, 'bold'))
        Nt.place(x=120, y=350)


Nt = tk.Label(win, text="Attendance filled Successfully", bg="Green", fg="white", width=40,
              height=2, font=('times', 19, 'bold'))
# Nt.place(x=120, y=350)

un = tk.Label(win, text="Enter username", width=15, height=2, fg="white", bg="blue2",
               font=('times', 15, ' bold '))
un.place(x=30, y=50)

pw = tk.Label(win, text="Enter password", width=15, height=2, fg="white", bg="blue2",
```

```
            font=('times', 15, ' bold '))
    pw.place(x=30, y=150)

    def c00():
        un_entr.delete(first=0, last=22)

    un_entr = tk.Entry(win, width=20, bg="yellow", fg="red", font=('times', 23, ' bold '))
    un_entr.place(x=290, y=55)

    def c11():
        pw_entr.delete(first=0, last=22)

    pw_entr = tk.Entry(win, width=20,show="*", bg="yellow", fg="red", font=('times', 23, ' bold '))
    pw_entr.place(x=290, y=155)

    c0 = tk.Button(win, text="Clear", command=c00, fg="black", bg="deep pink", width=10, height=1,
                       activebackground="Red", font=('times', 15, ' bold '))
    c0.place(x=690, y=55)

    c1 = tk.Button(win, text="Clear", command=c11, fg="black", bg="deep pink", width=10, height=1,
                activebackground="Red", font=('times', 15, ' bold '))
    c1.place(x=690, y=155)

    Login = tk.Button(win, text="LogIn", fg="black", bg="lime green", width=20,
                      height=2,
                      activebackground="Red",command=log_in, font=('times', 15, ' bold '))
    Login.place(x=290, y=250)
    win.mainloop()


###For train the model
def trainimg():
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    global detector
    detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
    try:
        global faces,Id
        faces, Id = getImagesAndLabels("TrainingImage")
    except Exception as e:
        l='please make "TrainingImage" folder & put Images'
        Notification.configure(text=l, bg="SpringGreen3", width=50, font=('times', 18, 'bold'))
        Notification.place(x=350, y=400)

    recognizer.train(faces, np.array(Id))
    try:
        recognizer.save("TrainingImageLabel/Trainner.yml")
```

```
    except Exception as e:
        q='Please make "TrainingImageLabel" folder'
        Notification.configure(text=q, bg="SpringGreen3", width=50, font=('times', 18, 'bold'))
        Notification.place(x=350, y=400)

    res = "Model Trained"  # +",".join(str(f) for f in Id)
    Notification.configure(text=res, bg="SpringGreen3", width=50, font=('times', 18, 'bold'))
    Notification.place(x=250, y=400)

def getImagesAndLabels(path):
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # create empth face list
    faceSamples = []
    # 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 = detector.detectMultiScale(imageNp)
        # If a face is there then append that in the list as well as Id of it
        for (x, y, w, h) in faces:
            faceSamples.append(imageNp[y:y + h, x:x + w])
            Ids.append(Id)
    return faceSamples, Ids


window.grid_rowconfigure(0, weight=1)
window.grid_columnconfigure(0, weight=1)
window.iconbitmap(r'C:/Users/Madhav Sharma/Desktop/FAMS/AMS.ico')

def on_closing():
    from tkinter import messagebox
    if messagebox.askokcancel("Quit", "Do you want to quit?"):
        window.destroy()


# def auto_mail_csv():
#     receiver = "madhav9620@gmail.com"  # receiver email address
#     body = "Attendence File"  # email body
```

```
#    filename =attach
#    # mail information
#    yag = yagmail.SMTP("frams.build@gmail.com",
#    receiver = "madhav9620@gmail.com"  # receiver email address
#    body = "Attendence File"  # email body
#    filename = attach # attach the file
#
#    # mail information
#    yag = yagmail.SMTP("frams.build@gmail.com", "azmyqgzabrvqhblb")
#
#    # sent the mail
#    yag.send(
#        to=receiver,
#        subject="Attendance Report",  # email subject
#        contents=body,  # email body
#        attachments=filename,  # file attached
#    )
# ")
#
#    # sent the mail
#    yag.send(
#        to=receiver,
#        subject="Attendance Report",  # email subject
#        contents=body,  # email body
#        attachments=filename,  # file attached
#    )


window.protocol("WM_DELETE_WINDOW", on_closing)

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

message.place(x=80, y=20)

Notification = tk.Label(window, text="All things good", bg="Green", fg="white", width=15,
                 height=3, font=('times', 17, 'bold'))

lbl = tk.Label(window, text="Enter Enrollment", width=20, height=2, fg="black", bg="deep pink",
font=('times', 15, ' bold '))
lbl.place(x=200, y=200)

def testVal(inStr,acttyp):
   if acttyp == '1': #insert
      if not inStr.isdigit():
```

```
        return False
    return True

txt = tk.Entry(window, validate="key", width=20, bg="yellow", fg="red", font=('times', 25, ' bold '))
txt['validatecommand'] = (txt.register(testVal),'%P','%d')
txt.place(x=550, y=210)

lbl2 = tk.Label(window, text="Enter Name", width=20, fg="black", bg="deep pink", height=2, font=('times', 15, ' bold '))
lbl2.place(x=200, y=300)

txt2 = tk.Entry(window, width=20, bg="yellow", fg="red", font=('times', 25, ' bold '))
txt2.place(x=550, y=310)

clearButton = tk.Button(window, text="Clear",command=clear,fg="black"  ,bg="deep pink"  ,width=10 ,height=1 ,activebackground = "Red" ,font=('times', 15, ' bold '))
clearButton.place(x=950, y=210)

clearButton1 = tk.Button(window, text="Clear",command=clear1,fg="black"  ,bg="deep pink"  ,width=10 ,height=1, activebackground = "Red" ,font=('times', 15, ' bold '))
clearButton1.place(x=950, y=310)

AP = tk.Button(window, text="Check Register students",command=admin_panel,fg="black"  ,bg="cyan"  ,width=19 ,height=1, activebackground = "Red" ,font=('times', 15, ' bold '))
AP.place(x=990, y=410)

takeImg = tk.Button(window, text="Take Images",command=take_img,fg="white"  ,bg="blue2"  ,width=20  ,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))
takeImg.place(x=90, y=500)

trainImg = tk.Button(window, text="Train Images",fg="black",command=trainimg ,bg="lawn green"  ,width=20  ,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))
trainImg.place(x=390, y=500)

FA = tk.Button(window, text="Automatic Attendace",fg="white",command=subjectchoose  ,bg="blue2"  ,width=20  ,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))
FA.place(x=690, y=500)

quitWindow = tk.Button(window, text="Manually Fill Attendance", command=manually_fill  ,fg="black"  ,bg="lawn green"  ,width=20  ,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))
quitWindow.place(x=990, y=500)

new = 1
url = "file:///C:/Users/Madhav%20Sharma/Desktop/FAMS/developers/diet1frame1first.html"
```

```
def openweb():
   webbrowser.open(url,new=new)

Btn = Button(window, text = "DEVELOPERS",command=openweb,fg="white" ,bg="reD"  ,width=20
,height=3, activebackground = "Red" ,font=('times', 15, ' bold '))
Btn.place(x=490, y=600)



window.mainloop()
```

4.1.1. Extra Features

In this project we add an extra feature called auto mail. It can automatically sent the attendance file to specific mail. Auto mail code given below:

Auto Mail

```
from tkinter import *
from tkinter import ttk

import smtplib
import webbrowser


def sendemail():
   try:

      sender = account.get()
      recepient = [receiver.get()]
      sub = subject.get()
      pswrd = password.get()
      msg = msgbody.get('1.0', 'end')
      msgtosend = """\
      From: %s
      To: %s
      Subject: %s
      %s
      """ % (sender, recepient, sub, msg)
      mail = smtplib.SMTP('smtp.gmail.com', 587)
      mail.starttls()
      mail.login(sender, pswrd)
      mail.sendmail(sender, recepient, msgtosend)
      mail.close()
```

```
        ttk.Label(mainframe, text="Email sent successfully").grid(column=4, row=9, sticky=W)

    except Exception as e:
        ttk.Label(mainframe, text=str(e)).grid(column=4, row=9, sticky=W)


def setup(event):
    webbrowser.open_new(r"https://www.google.com/settings/security/lesssecureapps")


root = Tk()
root.title("Send an Email via Gmail !!")

mainframe = ttk.Frame(root, padding="3 3 12 12")
mainframe.grid(column=0, row=0, sticky=(N, W, E, S))
mainframe.columnconfigure(0, weight=1)
mainframe.rowconfigure(0, weight=1)

account = StringVar()
password = StringVar()
receiver = StringVar()
subject = StringVar()
msgbody = StringVar()

a = Label(mainframe, text="To use this app turn this setting ON for your account", fg="blue",
cursor="hand2")
a.grid(columnspan=2, column=3, row=0, sticky=W)
a.bind("<Button-1>", setup)

ttk.Label(mainframe, text="Your Email Account: ").grid(column=0, row=1, sticky=W)
account_entry = ttk.Entry(mainframe, width=30, textvariable=account)
account_entry.grid(column=4, row=1, sticky=(W, E))

ttk.Label(mainframe, text="Your Password: ").grid(column=0, row=2, sticky=W)
password_entry = ttk.Entry(mainframe, show="*", width=30, textvariable=password)
password_entry.grid(column=4, row=2, sticky=(W, E))

ttk.Label(mainframe, text="Recepient's Email Account: ").grid(column=0, row=3, sticky=W)
receiver_entry = ttk.Entry(mainframe, width=30, textvariable=receiver)
receiver_entry.grid(column=4, row=3, sticky=(W, E))

ttk.Label(mainframe, text="Let's Compose").grid(column=2, row=5, sticky=W)

ttk.Label(mainframe, text="Subject: ").grid(column=0, row=6, sticky=W)
subject_entry = ttk.Entry(mainframe, width=30, textvariable=subject)
```

```
subject_entry.grid(column=4, row=6, sticky=(W, E))

ttk.Label(mainframe, text="Message Body: ").grid(column=0, row=7, sticky=W)
msgbody = Text(mainframe, width=30, height=10)
msgbody.grid(column=4, row=7, sticky=(W, E))

ttk.Button(mainframe, text="Send Email", command=sendemail).grid(column=4, row=8, sticky=E)

for child in mainframe.winfo_children(): child.grid_configure(padx=5, pady=5)

account_entry.focus()

root.mainloop()
```

Sample Images:

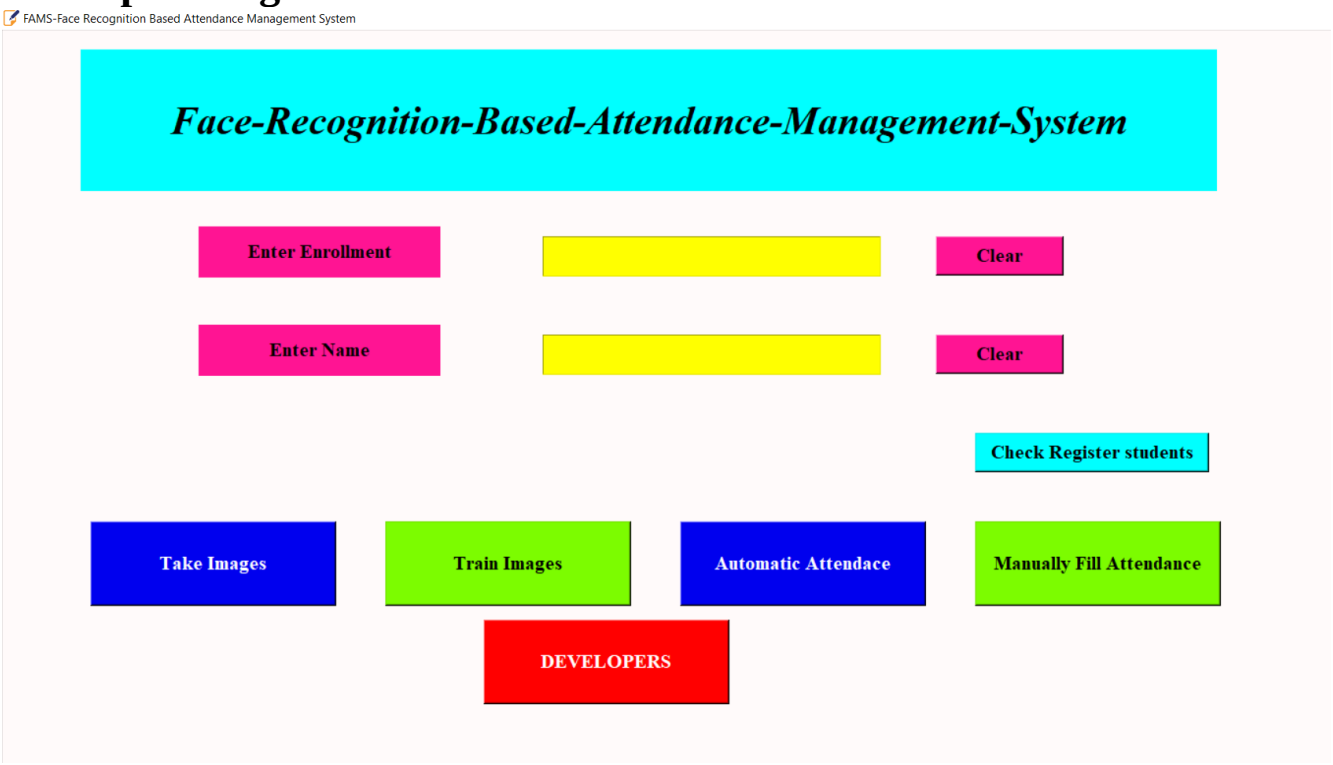
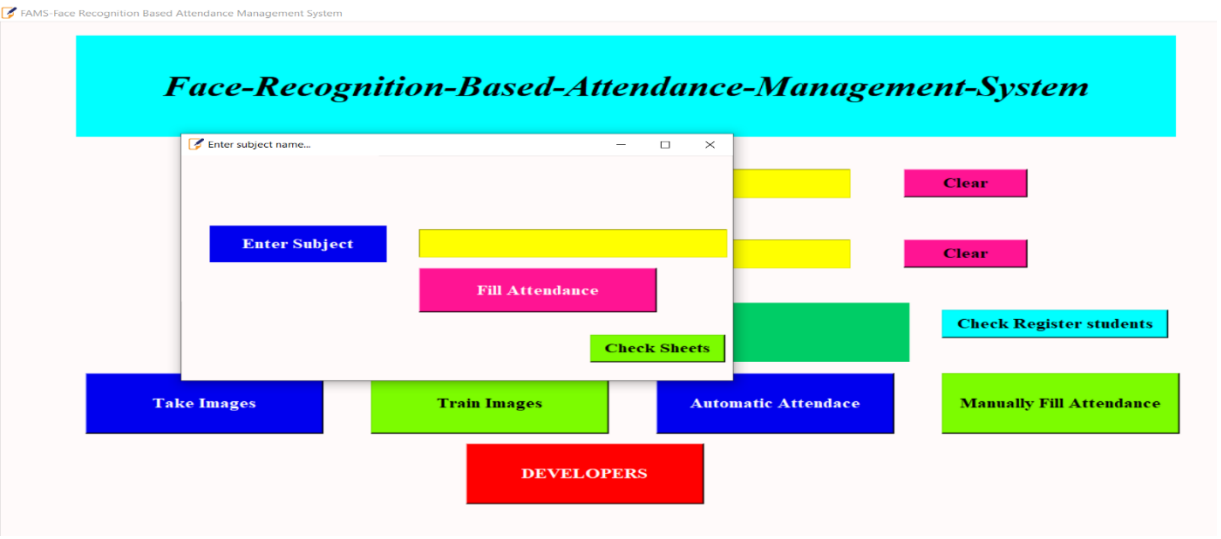
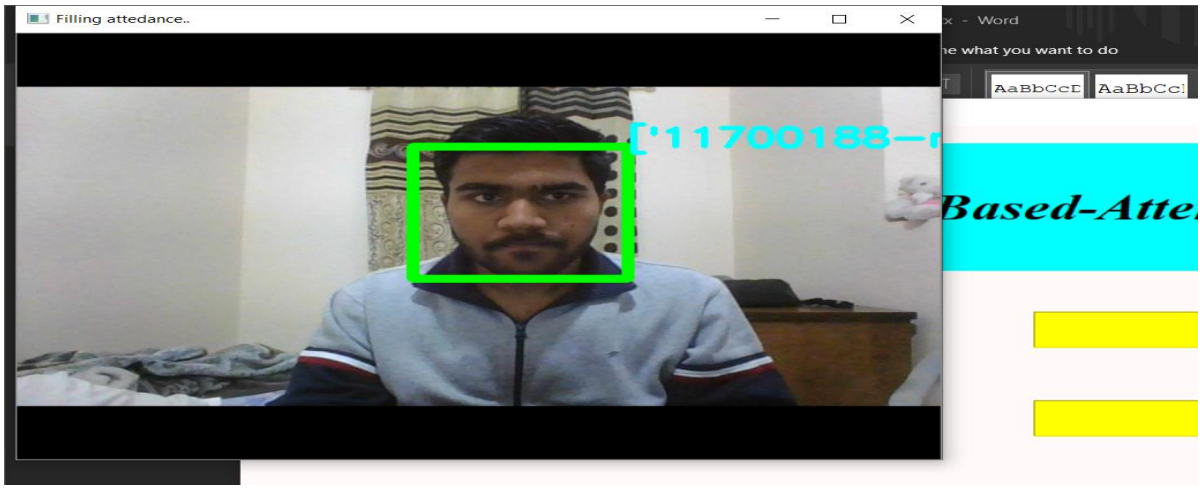


Figure : GUI of FAMS

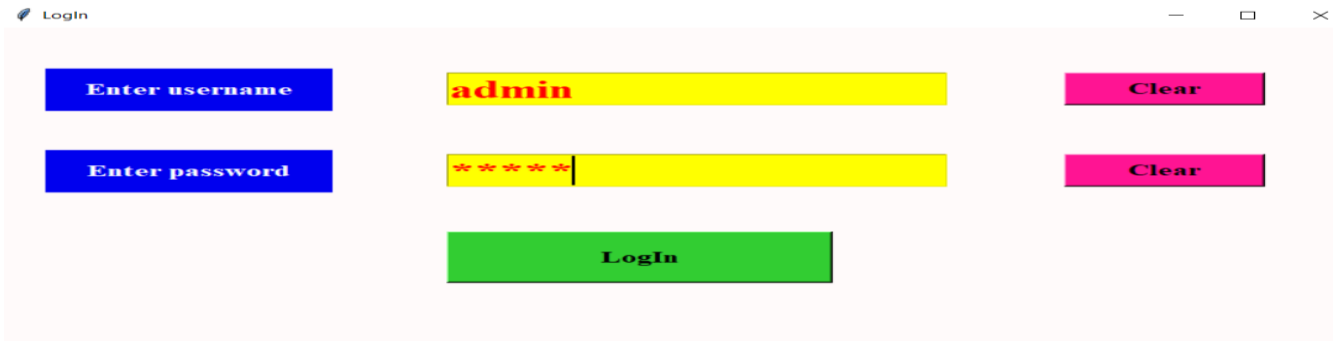
Taking Image For Dataset



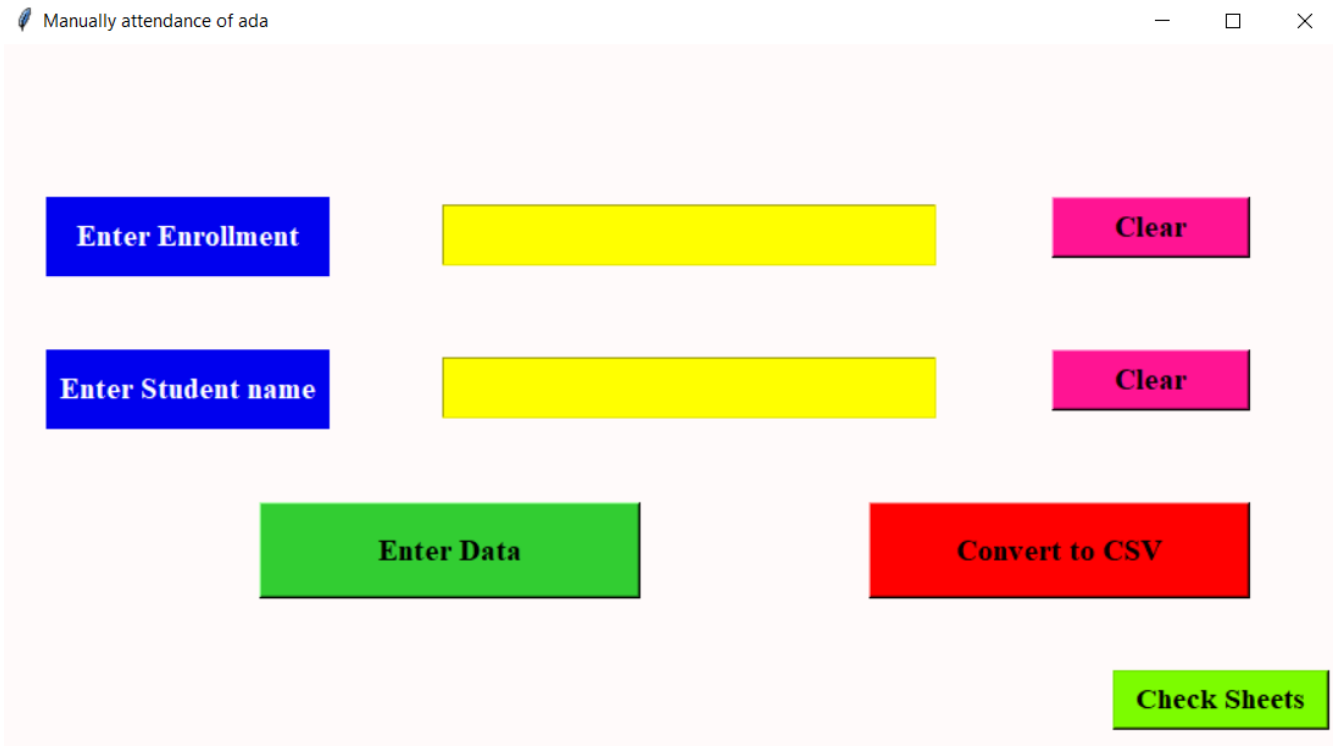
Automatic Attendance



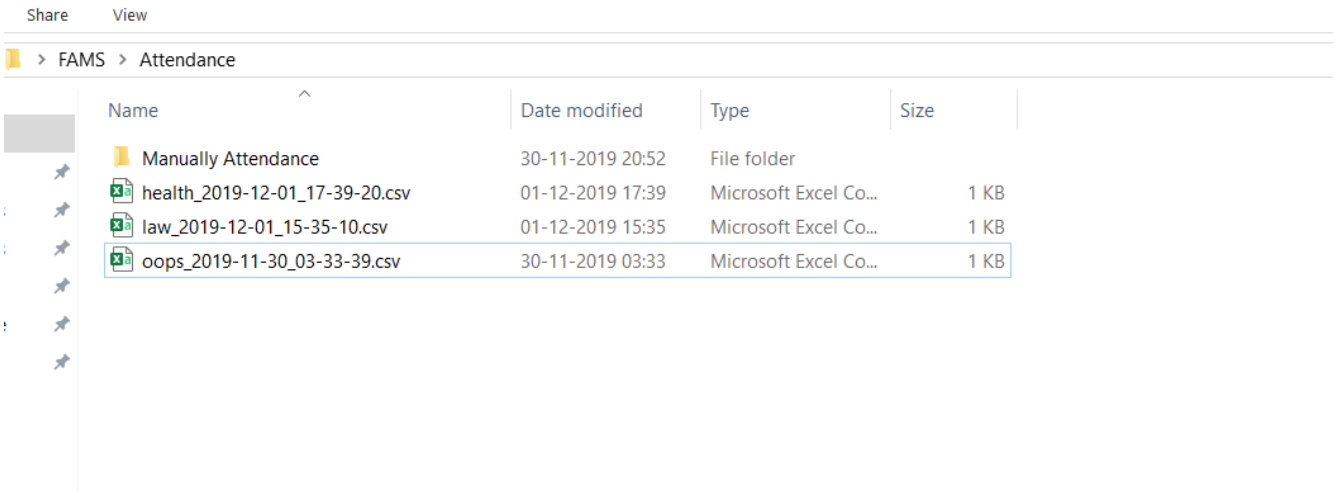
Face Recognition



Admin Panel



Manually Filling of Attendance



Attendance Marked Successfully

4.2. Summary

In this long yet useful chapter we managed to cover the entire structure of how the system has been developed and how it functions to give the best outcome.

Chapter 5

Working Plan

5.1. Introduction

In this chapter, we observe the entire work structure, meaning how the scheduling was maintained throughout the developmental phase and furthermore the feasibility study should be also discussed.

5.2. Work Breakdown Structure

In order to develop this system, we gave enormous importance to scheduling because we believed if we want to provide the best of quality in a given period of time then we must give due importance to scheduling which also helped us to achieve a better results. The figure below focuses the weekly work we had accomplished.

Week No.	Proposed Work
Week-1	Project Proposal Report and Presentation
Week-1	Study related works
Week-1	Study in Python
Week-1	Study related works using OpenCV
Week-1	Study related works using Bluetooth
Week-1	Study related works using processing
Week-2	Study image processing
Week-2	Study image processing
Week-2	Sketching basic structure
Week-3	Prototype design
Week-3	Finalize Prototype design
Week-3	Creating environment for image processing
Week-3	Creating environment for image processing
Week-3	Integrating all together
Week-4	Start coding
Week-4	Coding for basic instructions

Week-5	Coding for single face detection
Week-5	Single face detection and Compare with Database
Week-6	Multiple Face detection and Compare
Week-6	Detecting Multiple face, store and compare with database
Week-6	Attendance collection
Week-6	File Generate base on collective data
Week-6	Daily file generation of attendance

Table: Work Plan

5.3. Gantt Chart

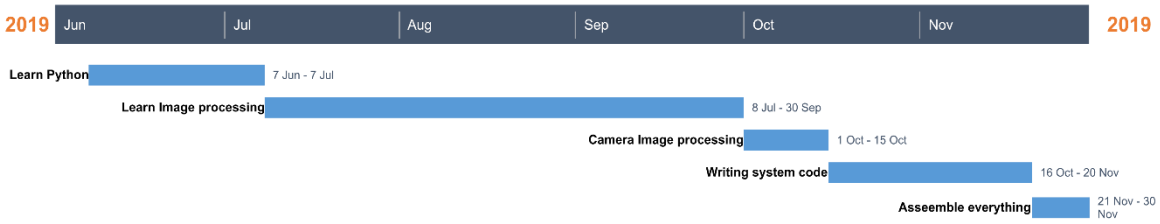


Figure : Gantt Chart

5.4. Feasibility Study

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. "**FEASIBILITY STUDY**" is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. It focuses on these major questions:

- What are the user's demonstrable needs and how does a candidate System meets them?

1. What resources are available for given candidate system?
2. What are the likely impacts of the candidate system on the organization?
3. Whether it is worth to solve the problem?

During feasibility analysis for on our project, following primary areas of interest are to be considered. Investigation and generating ideas about a new system does the following steps:

Steps in feasibility analysis

1. Form a project team and appoint a project leader.
2. Enumerate potential proposed system.
3. Define and identify characteristics of proposed system.
4. Determine and evaluate performance and cost effectively of each proposed system.
5. Weight system performance and cost data.
6. Select the best-proposed system.
7. Prepare and report final project directive to management.

Technical feasibility

A study of available resource that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not.

- Can the work for the project be done with current equipment existing software technology & available personal?
- Can the system be upgraded if developed?

- If new technology is needed then what can be developed? This is concerned with specifying equipment and software that will successfully satisfy the user requirement.

Economic feasibility

Economic justification is generally the "Bottom Line" consideration for most systems. Economic justification includes a broad range of concerns that includes cost benefit analysis. In this we weight the cost and the benefits associated with the candidate system and if it suits the basic purpose of the organization i.e. profit making, the project is making to the analysis and design phase.

The financial and the economic questions during the preliminary investigation are verified to estimate the following:

- The cost to conduct a full system investigation.
- The cost of hardware and software for the class of application being considered.
- The benefits in the form of reduced cost.
- The proposed system will give the minute information, as a result the performance is improved which in turn may be expected to provide increased profits.
- This feasibility checks whether the system can be developed with the available funds.

Operational Feasibility

It is mainly related to human organizations and political aspects. The points to be considered are:

- What changes will be brought with the system?
- What organization structures are disturbed?
- What new skills will be required?
- Do the existing staff members have these skills? If not, can they be trained in due course of time?

The system is operationally feasible as it very easy for the users to operate it.

Schedule feasibility

Time evaluation is the most important consideration in the development of project. The time schedule required for the developed of this project is very important since more development time effect machine time, cost and cause delay in the development of other systems.

5.5. Summary

To conclude, we discussed the scheduling processes of developing this system. Additionally we have also identified how feasible the system is through the lens of evaluating using various feasibility studies

Chapter 6

Future Work

6.1. Introduction

This chapter discusses the future scope or the implementation of this robot. To incerease the scope of this device we can add some new features. As technology is becoming more advance it will be mendatory to change the sturctute some day with better replacement and sometimes based on customer requirements.

6.2. Future Scope of Work

There are so many future scope on this project. Some of them are

- Can improve security
- Can use Neural Network for high accuracy
- Can used in big factory or employee attendance
- Can build on fully web base system.

6.3. Summary

This chapter has described the possible future applications of the design. But there are a lot of possibilities with the designed device. The device may need some research for different applications, though the principle of the designed system will remain as it is.

Chapter 7

Result

7.1. Introduction

This chapter of the report contains the results that we achieved throughout the course of using this system.

Results Achieved From initiation through conclusion of developing this system the following results has been achieved. They are as follows:

- The system can be administered by a non-IT technician.
- The system is market ready for commercial use.
- The system has the capacity to carry up to a thousand faces to recognize.
- The system can serve as much people as they want within an organization.

7.2. Summary

This chapter has covered the different types of results that we have managed to obtain throughout the course of using this system.

Chapter 8
Refrences

1. Edureka blog for machine learning, https://www.edureka.co/blog/what-is-machine-learning/
2. OpenCv Documentation for Face Recogniser, https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html
3. OpenCv Documentation for Face Recogniser, https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#introduction
4. OpenCv Documentation for Face Recogniser, https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#face-recognition
5. OpenCv Documentation for Face Recogniser, https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#face-database
6. OpenCv Documentation for Face Recogniser, https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#eigenfaces
7. OpenCv Documentation for Face Recogniser, https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#fisherfaces
8. OpenCv Documentation for Face Recogniser, https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms
9. Open Cv Documentation for image processing
10. https://docs.opencv.org/3.0beta/doc/py_tutorials/py_gui/py_video_display/py_video_display.html