Deep Unified Model For Face Recognition Based on Convolution Neural Network

A project report submitted in partial fulfillment of the requirements

for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By

A. ARAVIND	**17HU1A0503**
B. RAJITHA	**17HU1A0533**
D. ANITHA	**17HU1A0501**
CH. SILPA	**17HU1A0542**
V. LOKANATH GANGULI	**17HU1A0523**

Under the esteemed guidance of

Mr. K. MANIKANTA **M.Tech**,

Assistant Professor

DEPARTMENT

OF

COMPUTER SCIENCE & ENGINEERING

TADIPATRI ENGINEERING COLLEGE

Approved by AICTE and Affliated to JNTUA, Kadapa Road, Veerapuram Village, Tadipatri, Andhra Pradesh-515411.

2017-2021

TADIPATRI ENGINEERING COLLEGE

Kadapa Road, Veerapuram (V), Tadipatri, Andhra Pradesh-515411.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that this project work entitled **"Deep Unified Model For Face Recognition Based on Convolution Neural Network"** is being submitted by **A.ARAVIND (17HU1A0503), B.RAJITHA (17HU1A0533), CH.SILPA (17HU1A0542), D.ANITHA (17HU1A0501), V. LOKANATH GANGULI (17HU1A0523)**, in partial fulfillment of the requirements for the award of the graduation degree of **Bachelor of Technology** in COMPUTER SCIENCE & ENGINEERING from the Jawaharlal Nehru Technological University,Anantapuram during the period of 2017-2021 is a record of bonafide work carried out by us under our esteemed guidance and supervision**.** The result provided in this report has not been submitted to any other University or Institution for the award of any degree.

INTERNAL GUIDE

Mr. K. MANIKANTA M.Tech
Assistant Professor,

Department of CSE

Tadipatri Engineering College

HEAD OF THE DEPARTMENT

Mr .C. NAGESH M.Tech, (Ph.D)
Associate Professor

Department of CSE

Tadipatri Engineering College

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the work is being presented in this dissertation entitled **"Deep Unified Model For Face Recognition Based on Convolution Neural Network"** submitted towards the partial fulfillment of requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering, work carried out under the supervision of **Mr.K.MANIKANTA M.Tech, Asst Professor, TADIPATRI ENGINEERING COLLEGE, Tadipatri.** The results embodied in this dissertation report have not been submitted by me for the award of any other degree. Further more, the technical details furnished in various chapters of this report are purely relevant to the above project and there is no deviation from the theoretical point of view for design, development and implementation.

A.ARAVIND	**17HU1A0503**
B.RAJITHA	**17HU1A0533**
D.ANITHA	**17HU1A0501**
CH.SILPA	**17HU1A0542**
V.LOKANATH GANGULI	**17HU1A0523**

ACKNOWLEDGEMENT

We are here by acknowledge our profound gratitude to our beloved chairman, **Sri K.Ramesh Reddy B.Sc, B.L**, who took keen interest and encouraged us in every effort throughout the course.

We are our gratitude to our Principal **Dr. EV Subba Reddy Ph.D,** for his kind attention in the valuable guidance to us throughout the course.

It would be honor to thank our Vice principal **Mr. M.C Bhanu Prasad M.Tech,(Ph.D)**,for his support and encouragement.

We are expressing heart-full gratitude and deep indebtedness to beloved Professor **Mr. C.Nagesh M.Tech,(Ph.D)**, Head of the Computer Science and Engineering, **Tadipatri Engineering College**, for his great help and encouragement in doing my main project successfully.

We sincerely thank our guide **Mr. K. Manikanta M.Tech,** Asst. Professor, for his guidance and valuable suggestions and support in the completion of the project.

Finally, we thank all the faculty members, supporting staff of CSE dept, and our friends for their kind co-operation and valuable help for completing the project.

ABSTRACT

Face recognition is of great importance to real world applications such as video surveillance, human machine interaction and security systems. As compared to traditional machine learning approaches, deep learning based methods have shown better performances in terms of accuracy and speed of processing in image recognition. This paper proposes a modified Convolutional Neural Network (CNN) architecture by adding two normalization operations to two of the layers. The normalization operation which is batch normalization provided accelerating the network. CNN architecture was employed to extract distinctive face features and Navi Bayes, Softmax classifier was used to classify faces in the fully connected layer of CNN. In the experiment part, Georgia Tech Database showed that the proposed approach has improved the face recognition performance with better recognition results.

INDEX

LIST OF FIGURES

LIST OF TABLES

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 enrollment 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.

Robinson - Riegler,G., & Robinson- Riegler, B. (2008) mentioned that 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. 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.

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 (Robert Silk, 2017). 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).

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.

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.

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.

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 Tools to be used in this project

Tools will be described into following:

- **Python**

Python is high-level programming language like a Perl, Ruby and Tcl which are used as a scripting language. It was conceived by Guido van Rossum in 1989. Also, python is one of the three "official languages" in Google which means that more application in Google was deployed this language. E.g. Google App Engine SDK. Here are the points that make python is selected:

1. Free, Python is product of open source. People allows to use it in business or commercial without any charge.

2. Easy to read, Syntax in Python is clear and readable. Beginner can be easily to read and handle Python's coding very well.

3. Rapid development, it is because it likes pseudo code. Everything coding in Python is direct result.

4. Highly portability, Python is working on different platforms, because of Python is written portable ANSI C.

5. Reusability, Python is easily reused modules and packages. Peoples can be developed their own library and reused it later project.

6. Object-Oriented Programming. Unlike scripting language, Python is designed to be object-oriented. OO programming means you can implemented using idea of inheritance and polymorphism.

- **OpenCV**

OpenCV is a synonym of Open Computer Vision Library, which has at least 500. algorithms, documentation and sample code for real time computer vision. OpenCV is originally developed by Intel and launched in 1999. It free for commercial and research used. OpenCV library is cross-platform which means its can execute on Windows, Mac OS X, Linux, PSP and other embedded devices.

The library is mainly written in C, which makes it easily possible to transfer into specific platforms. Example application of OpenCV library is Human-Computer Interaction, Object Identification, Segmentation and Recognition and so on. Stanley was implemented by OpenCV, which was the winning entry to the 2005 DARPA Grand Challenge race.

1.5 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.

1.6 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.

1.7 Flow Chart

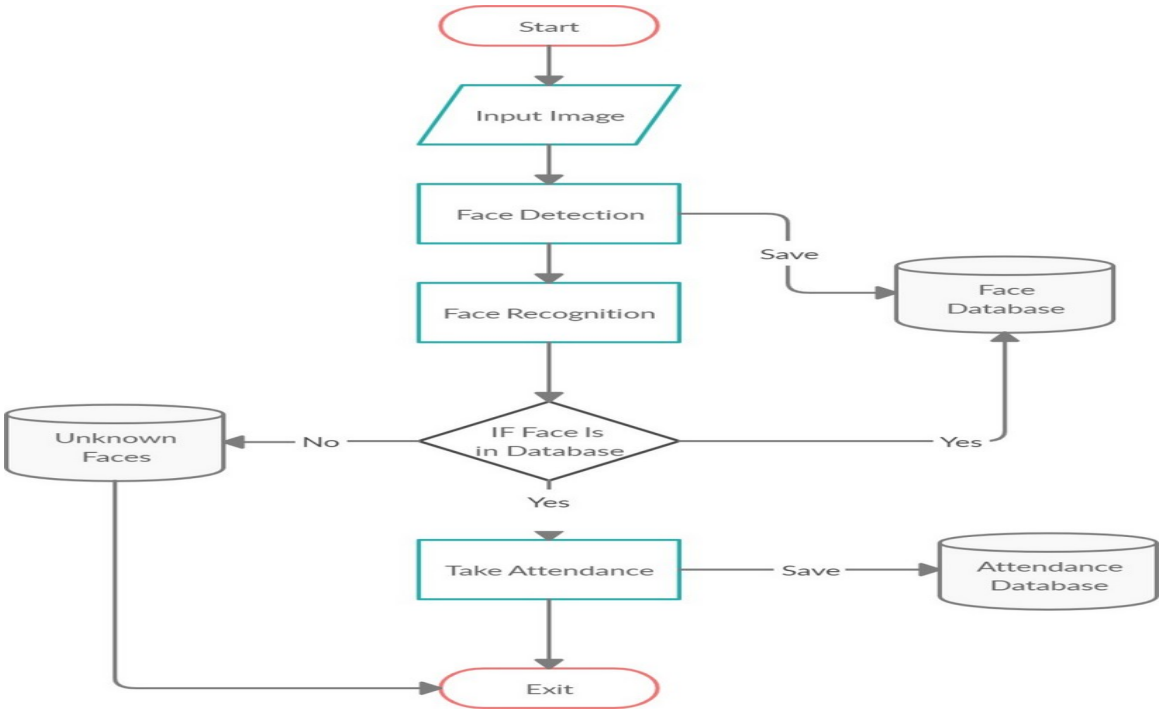


Fig 1.1: Image Detection

1.8 Summary

This project is presented the background of computer vision to address the important of face detection in our real life. Objective of project is being presented and let the people know the basic requirement of hardware and software which tools are used to implement.

CHAPTER 2

LITERATURE SURVEY

2.1 Student Attendance System

Arun Katara et al. (2017) mentioned 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. 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	Accurate	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

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

2.3 Image Representation in a Digital Computer

An image is a 2-Dimensional light intensity function $\mathbf{f}(\mathbf{x},\mathbf{y}) = \mathbf{r}(\mathbf{x},\mathbf{y}) \times \mathbf{i}(\mathbf{x},\mathbf{y})$ - (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. 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 256×256, elements, 640×480 pels or 1024×1024 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).

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.
- Pre processing – 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.
- Knowledge Base – This helps for efficient processing as well as inter module cooperation.

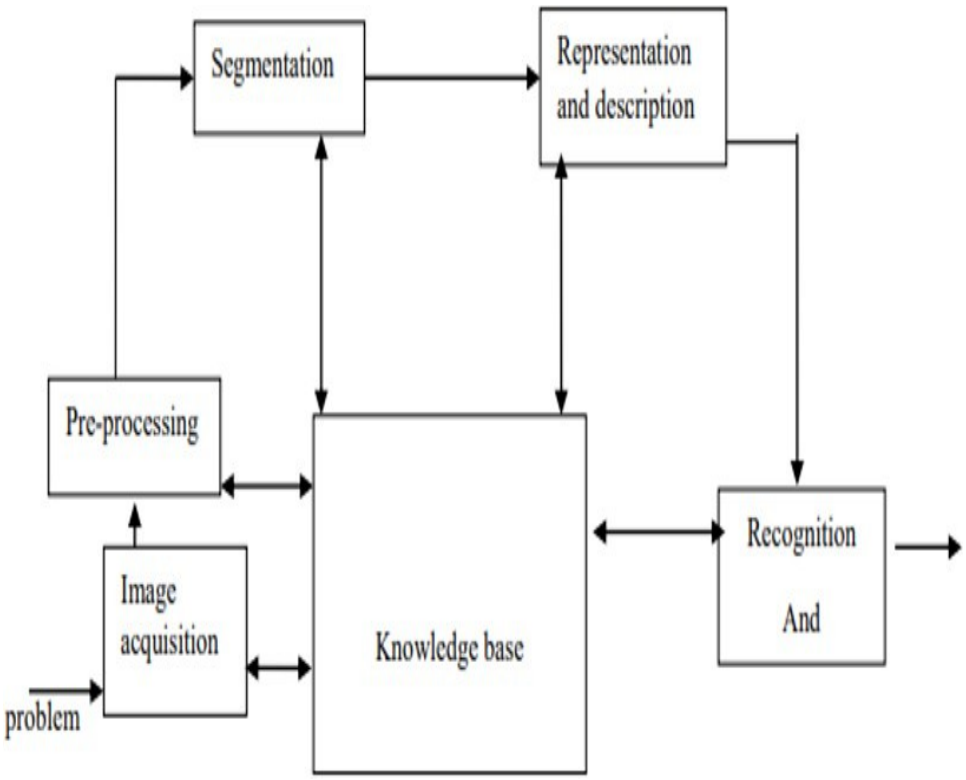


Fig 2.1 :Steps in Digital Image processing

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.

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/non face 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	1.Only used for binary and grey images. 2.Overall performance is inaccurate compared to Viola-Jones Algorithm.

Face Detection Method	Advantages	Disadvantages
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.

Table 2.2 : Advantages & Disadvantages of Face Detection Methods 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.

Fig 2.2: Haar Feature

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.

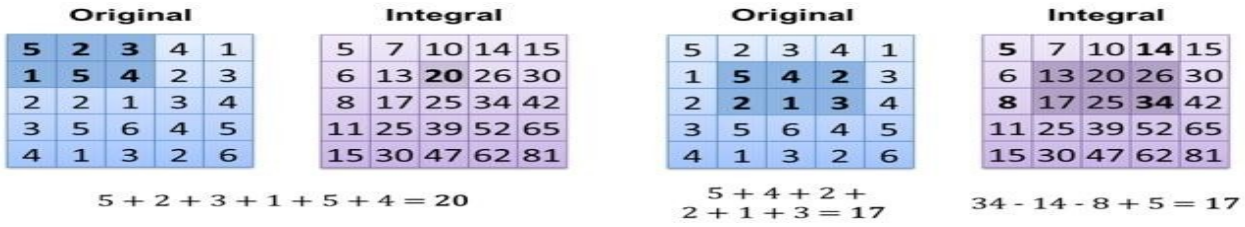


Fig -2.3 : Integral of Image

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.

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:** 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.

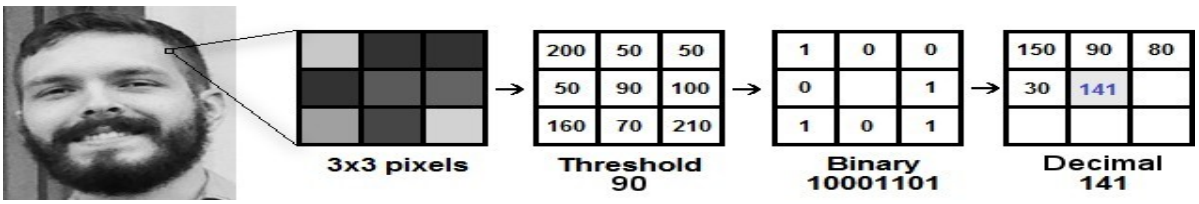


Fig – 2.4: LBP Operation

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

- 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.

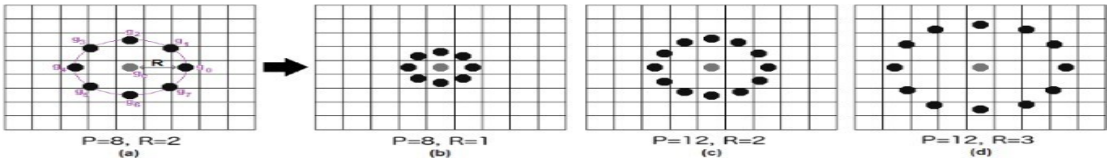


Fig 2.5: The LBP operation Radius Change

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.

4 **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:

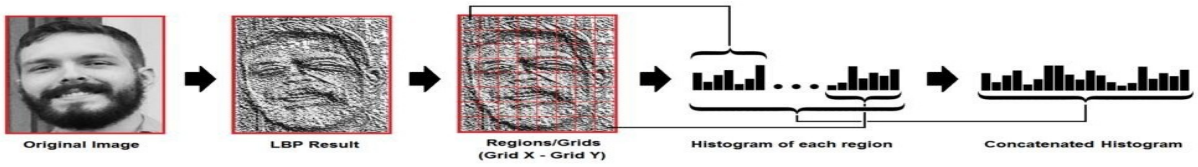


Fig 2.6 : Extracting The Histogram

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. final histogram represents the characteristics of the image original image.

5 **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.

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

- 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: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.

2.6 Method Approaches for face detection

In general, FD can be implemented by four methods: knowledge based methods, template matching, invariant feature methods and learning based methods. These methods will be introduced with the following:

1. Knowledge based methods

The models are used human knowledge to find a face patterns from the testing images. Based on the nature of human faces, algorithms scan the image from top-to-bottom and left-to-right order to find facial feature. For instance, face should be including two eyes and mouth…

Pros: Easy applied into simple rules

Cons: difficult to detect in invariant background, such as different pose, uncontrolled illumination. Well results based on well-defined rules. This algorithm does not work on the pose.

2. Template marching

The model is used several templates to find out the face class and extract facial features. Rules are pre-defined and decide whether there is face in the image. For instance, using filters to extract the contours of face shape Sample of template marching

Pros: Simple to apply this method.

Cons: similar to knowledge based method, hard to detect face in different poses.

Algorithms are sensitive to scale size, face shape and pose.

Image-based Face Detection System

3. **Invariant feature methods**

The model is bottom-up approaches and used to find a facial feature (eyebrows, nose), even in the presence of composition, perspective vary, so it is difficult to find a face real time using this method. Statistical models are developed to determine the faces. Facial features of human

faces are: shape, texture, skin.

Pros: Unlike knowledge-based method, it is invariant to pose and expression.

Cons: not suitable to detect facial features from uncontrolled background, time

consuming algorithms. Detection rate is not accuracy, because of need to combine different feature and processing it.

4. **Learning based methods**

The models are trained from a set of training set before doing detection. For the large amount of training data, it can be provided high accuracy recognition rate to resist variation, expression and pose of faces images. For instance, Many of “non-face” and “face” images import into the system. Machine learning techniques are employed to train the system based on the statistical properties and probability distribution function. Principle Component Analysis (PCA), Support Vector Machine (SVM), Naïve Bayes Classifier, Hidden Markov model, Neural Network and Adaboost are well-known classifiers to use for face detection.

Pros: fast to detect face. Can be detected different pose and orientation if have enough training set. Showed a good empirical results.

Cons: need more and more “non-face” and “face” sample for training, need to scan different scale.

Image-based Face Detection System

- **Global and component methods**

Above 4 methods can be grouped into two categories: global and component. In global method, every feature vector that represents a whole face image. A Major problem in global method is variation in the appearance of object. However, in component method, faces are extracted and become facial components and combine them into single feature vector which is classified by classifier. These setting will be increased the processing time and resource.Example of global approaches face detection Example of component-based approaches face detection.

2.7 Summary

This project is presented about the introduction of face detection, face database, the major challenges of face detection. In addition, four face detection methods are introduced to give general ideal to user about the approaches of research area. Also, the performance evaluation is addressed to show how is the successful detection system should be achieved.

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 Pre-processing

System input is color images which included images of human faces or not, output is the human faces which is extracted from original images. In order to get the better result of detection, pre-processing is essential. In this section, pre- processing is addressed with giving detail description.

Gray scale conversion

For getting to reduce the information of images, image should be done a converting to grayscale. Each color images (RGB images) are composed of 3 channels to present red, green and blue components in RGB space. Below is the example to giving the general ideal of the RGB color image. Image is defined with grayscale level, which means pixels in images are stored 8-bit integer to represent color from black to white

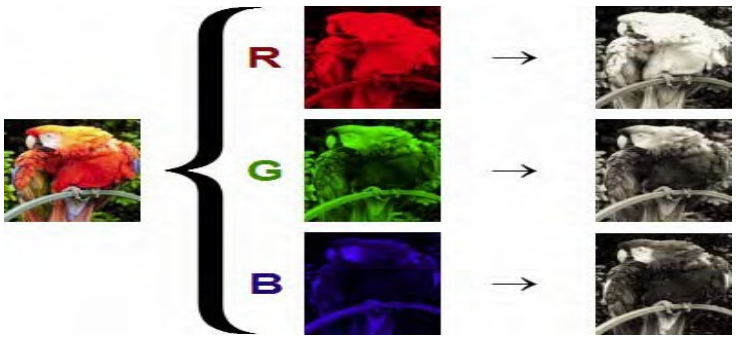
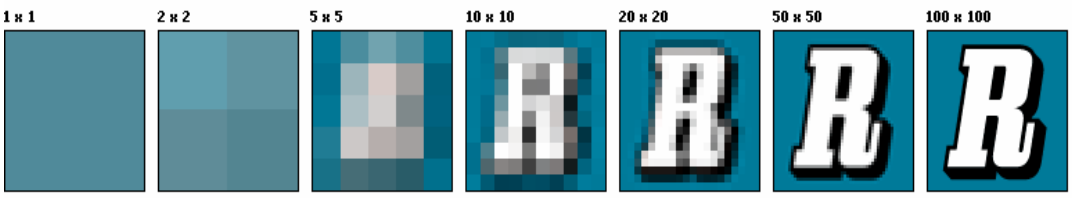


Fig 3.1 : Image Gray scale Conversion

- **Convert to Gray scale Image algorithms**
  - Given example images (R_1,G_1,B_1),….(R_n,G_n,B_n) where R, G, B are the value of red, green and blue respectively and ‘n’ is total number of pixel in given image.
  - The new grayscale images has pixel from G_1,…..G_n, where using formula is as follows: 0.21R + 0.71G + 0.07 B = G . Unlike averages method, this form is considering the ratio because of human perception.
  - Image resizing

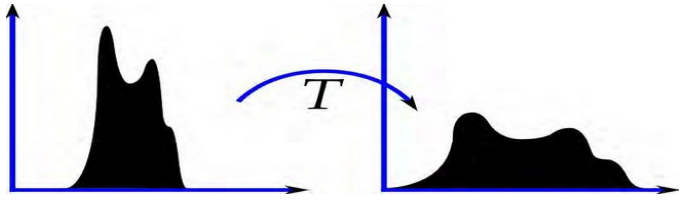
Images are synthesized by numerous of pixel which is the small unit in the image. Also, images are the 2-dimensional matrix pattern, every pixel in the images is represented something information. For example, ‘0’ is white and ‘255’ is black in gray scale images. Because there are a lot of information to deal with, input images are employed the resizing processing to reduce the images resolution with keeping same quantity. Below example is an illustration about the different resolution to describe the same image.



The top-left side of each image is the resolution of each one. Left-side’s image is the original.Image has 3000 pixels in width and 2000 pixels in height which means it has 3000 x 2000 = 6,000,000 pixels or 6 megapixels. If the image has been resized into 1000 pixels in width and 600 pixels in height, image only has 0.6 megapixels. At least system only use 1/10 timing to handle it. However, if you had to adjust the size of the image will also affect the face detection rate.

Histogram Equalization

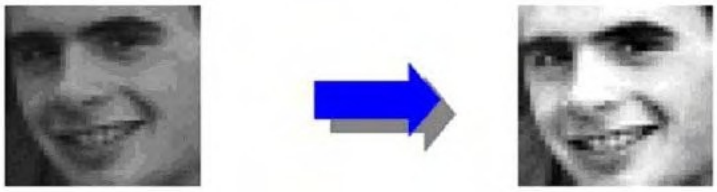
Histogram equalization is a statistical method of images processing. It works as a statistical histogram of color distribution of the average scattered in the histogram, so that the distribution of a histogram graph homogenization. The ideal is present as a follows:



change of the histogram after perform histogram equalization

In the above chart, the shape of graph has been widened which is the meaning of average scattered in the histogram.

This method usually increases the contrast of the input images. In face detection system, The left-hand side of below images is resized grayscale images. Other is output images after proceed the processing of histogram equalization. You will see very significant results.



Example of the process of histogram equalization

- **Algorithms of Histogram equalization**

Grayscale images has Xn pixels with i represent a value of gray level in each pixel.The following chart is represent the relationship between probability of occurrence and the value of each pixel:

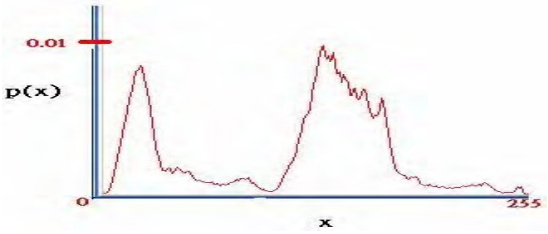


Chart of Probability Density Function (PDF) And being histogram of images and normalized to [0,1] define the cumulative distribution function as follows:

$$F(n) = \sum_{i=0}^{n} p(x_i) \qquad \sum_{i=0}^{255} p(x_i) = 1 \qquad n = 0,1,2...255$$

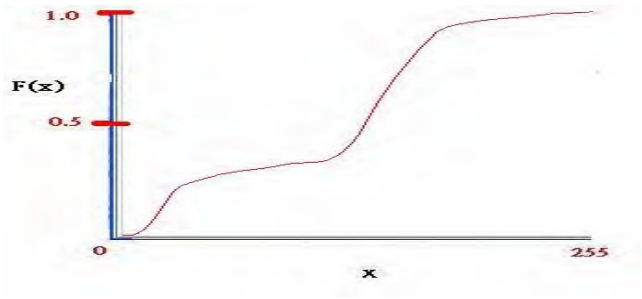


Chart of Cumulative distribution function

Minimum and maximum value are found and applied into following equation to find out the histogram equalization of each pixel:

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$

Where cdf min is the minimum value of CDF, M is the width of image and N is the height of image. L represent a large value of grey level, = 256.

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 ca. mera 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.

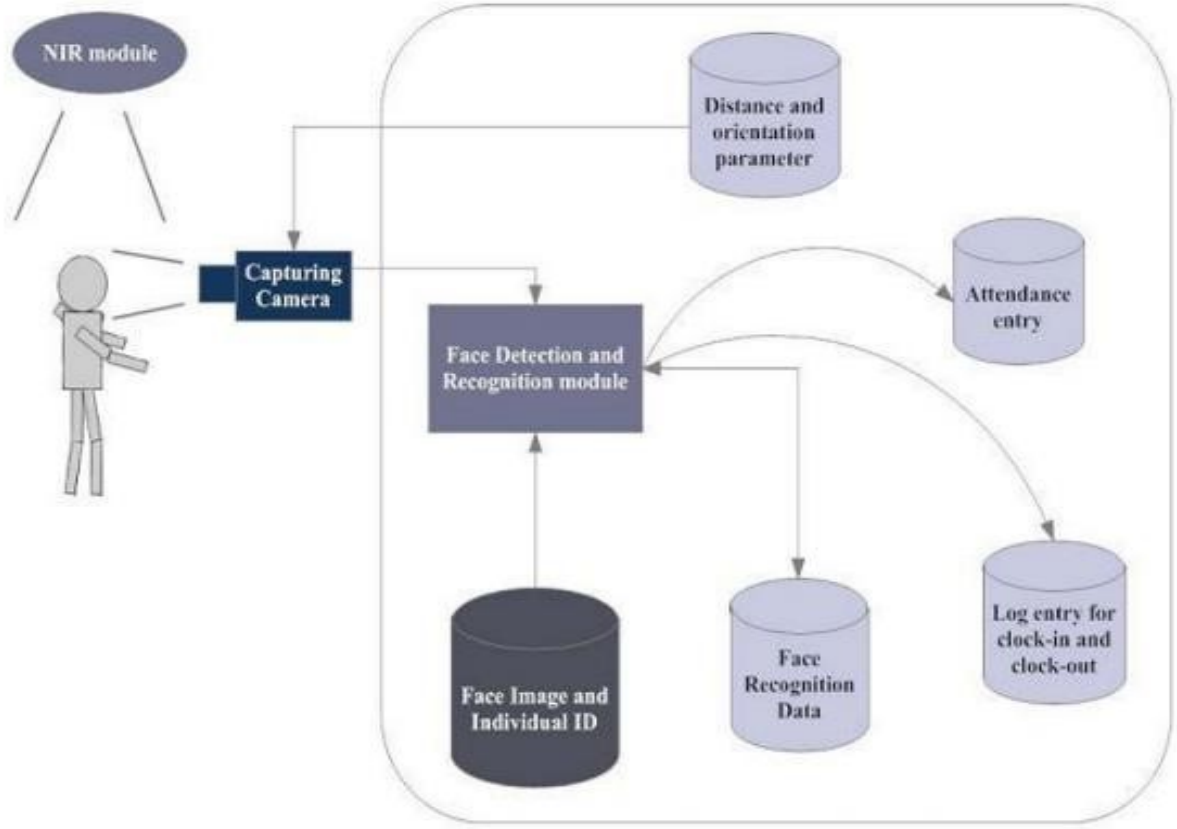


Fig 3.2 : Data flow diagram

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
- **Interest points**: detection and matching
- **Video processing:** optical flow, background subtraction, camshaft (object tracking)
- **Photography**: panoramas realization, high definition imaging (HDR), image inpainting.

So it was very important to install OpenCV. But installing OpenCV 3 is a complex process.

```
#!/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 [ "$(id -u)" != "0" ]; then
   echo "This script must be run as root" 1>&2
   exit 1
fi
echo installing core dependencies ...
apt-get -y install cmake python3-dev python3.4-dev python3-numpy gcc build-essential cmake-curses-gui
echo installing other dependencies ...
apt-get -y install pkg-config libpng12-0 libpng12-dev libpng++-dev libpng3 libpnglite-dev zliblg-dbg zliblg
zliblg-dev pngtools libtiff5-dev libtiff5 libtiffxx0c2 libtiff-tools libeigen3-dev
echo installing helper apps ...
apt-get -y libav-tools
#apt-get -y ffmpeg libavcodec55 libavformat55
apt-get -y install libjpeg8 libjpeg8-dev libjpeg8-dbg libjpeg-progs libavcodec-dev libavformat-dev libgstreamer0.10-0-dbg
libgstreamer0.10-0 libgstreamer0.10-dev libxine2-ffmpeg libxine2-dev libxine2-bin libunicap2 libunicap2-dev swig libv4l-0 libv4l-dev libpython3.4 libgtk2.0-dev
echo Receving OpenCV 3.0.0 source...
git clone --branch 3.0.0 --depth 1 https://github.com/Itseez/opencv.git
cd opencv
mkdir release
cd release
echo Preparing compilation, may take a long while...
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=$(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 !
```

Fig 3.3 : Installing OpenCV

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.

These are the commands we used.

```
1. Sudo chmod 755 /myfile/pi/installopencv.bash
2. sudo /myfile/pi/installopencv.bash
3.py -m pip install opencv-python
```

2.Python IDE: There are lots of IDEs for python. Some of them are PyCharm, Thonny, Ninja, Spyder etc. Ninja and Spyder both are very excellent and free but we used Spyder as it feature- rich than ninja. Spyder is a little bit heavier than ninja but still much lighter than PyCharm. You can run them in pi and get GUI on your PC through ssh-Y. We installed Spyder through the command line below.

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

3.3.2.Hardware Implementation

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 8GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data

1.HD Webcam:

ELP HD 8Megapixel USB CMOS board camera module adopt Sensor Sony (1/3.2”) IMX179 is nice to use in Linux equipment, or those equipment which come with windows, linux, Android system.

Fig 3.4 : Webcam

- **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**

Using PCA algorithm the following steps would be followed in for face recognition:

Begin:

- update the log table with corresponding face image and system time that makes completion of attendance for an individual students.

End :

This section presents the results of the experiment conducted to capture the face into a grey scale image of 50x50 pixels.

Test data	Expected Result	Observed Result	Pass/ Fail
OpenCAM_CB()	Connects with the installed camera and starts playing.	Camera started.	pass
LoadHaar Classifier()	Loads the HaarClassifier Cascade files for frontal face	Gets ready for Extraction.	Pass
ExtractFace()	Initiates the Paul-Viola Face extracting Frame work.	Face extracted	Pass
Learn()	Start the PCA Algorithm	Updates the facedata. xml	Pass
Recognize()	It compares the input face with the saved faces.	Nearest face	Pass

Table 3.1 :Methods of Face Recognition

our data set sample.

Fig 3.5: Dataset sample

Face Orientations	Detection Rate	Recognition Rate
O^0 (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.

3.4 Summary

This section present the methodology used in the project. They are included two parts. The first part is pre- processing section, which can be divided into “Gray scale Conversion”, “Image Resizing” and “Histogram Equalization”. Second part is algorithm section, which are “Integral Image”, “AdaBoost” and “Cascade”.

CHAPTER - 4

SYSTEM DESIGN

This section gives a list of system requirement and more detail information on the application of this project. It is named as FDS (Face Detection System).

4.1 Overview

The core ideal of the Face Detection System (FDS) is illustrated as below picture:

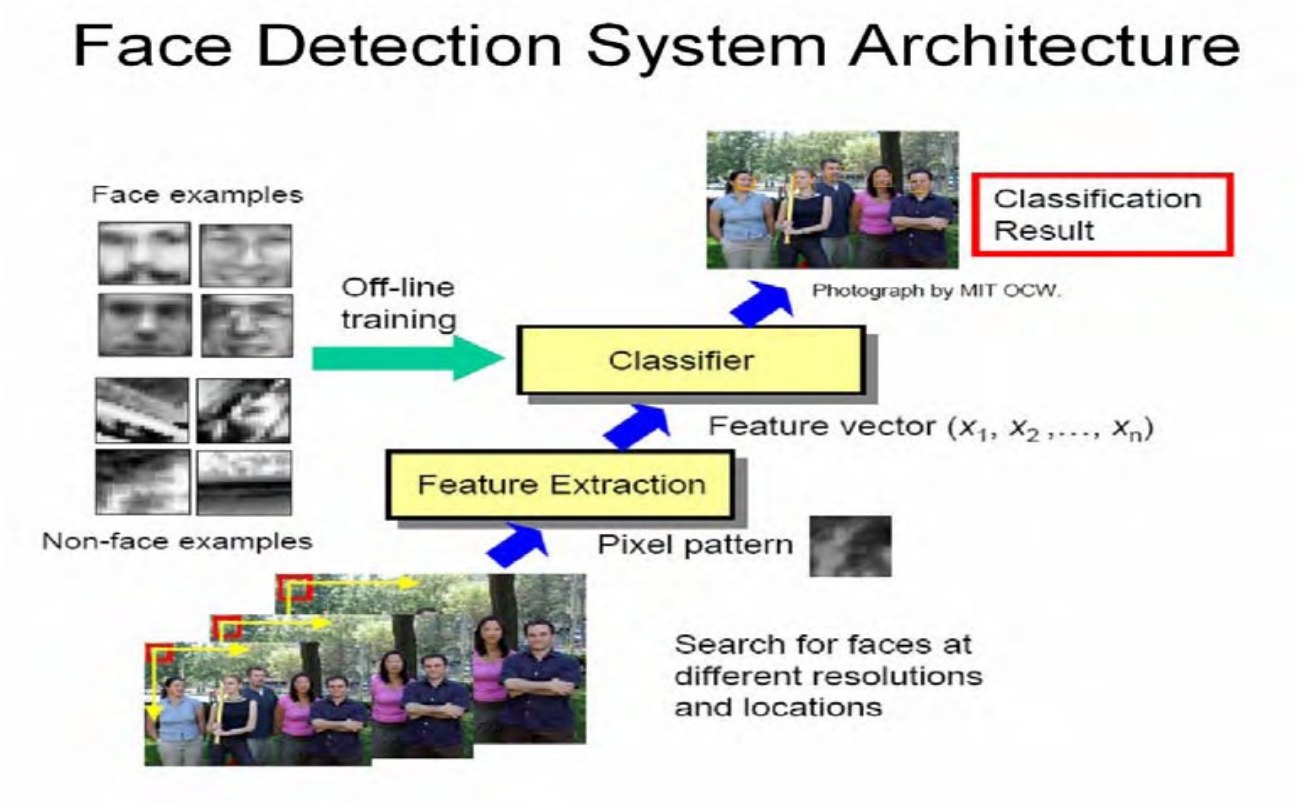


Fig 4.1 :Face detection system Architecture

4.2 Functional Requirements

The following list is a basic requirement of the FDS:

1 Support Face Detection

2 Label each detected face in the Picture

3 Support select images from other folder

4 Show clear images list to user

5 Display the number of ‘detected’ faces

4.3 System Structure – UML

The following sections are the list of diagrams which are use case, activity diagram, class diagram.

Use Case Descriptions

Figure is overview of use case in FDS. These use cases are described more detail in specific condition.

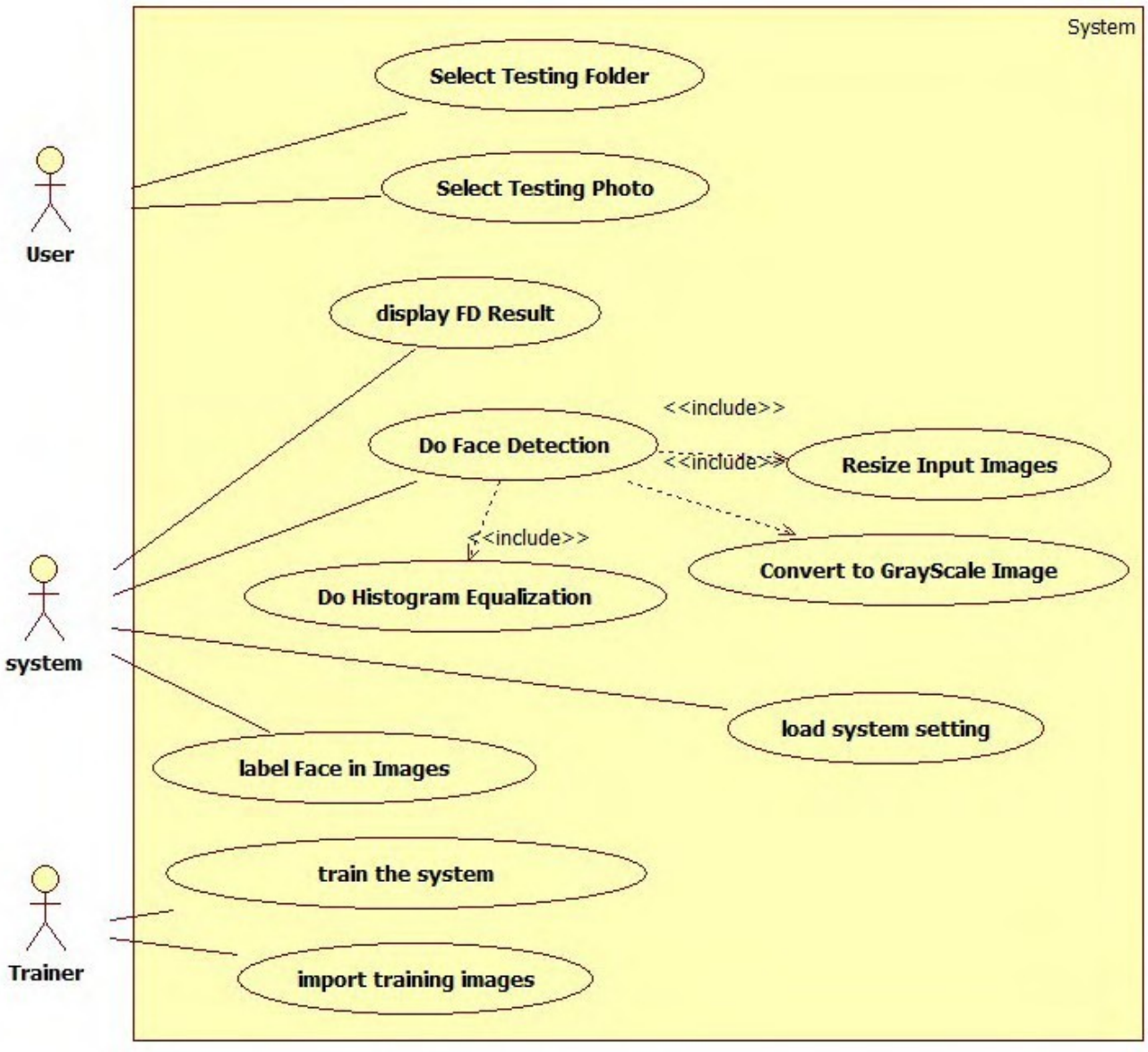


Fig 4.2:Use case Diagram

Sequence Diagram – Face detection

The following diagram is face detection's sequence diagram which is core section in the system.

1 The class of File Handle is prepared the list of the filenames that are selected by User and displayed in the User interface.

2 User select specific "image" to perform face detection

3 FDS call and create new object from FD Modules.

4 Input image is being to experience resize processing.

5 Input image is being to convert grey scale mode.

6 Input image is being to perform histogram equalization.

7 Done Message will be sent out to FD Modules.

8 Pre- processing is finished and performs face detection.

9 The result of face detection is shown in FDS's User Interface.

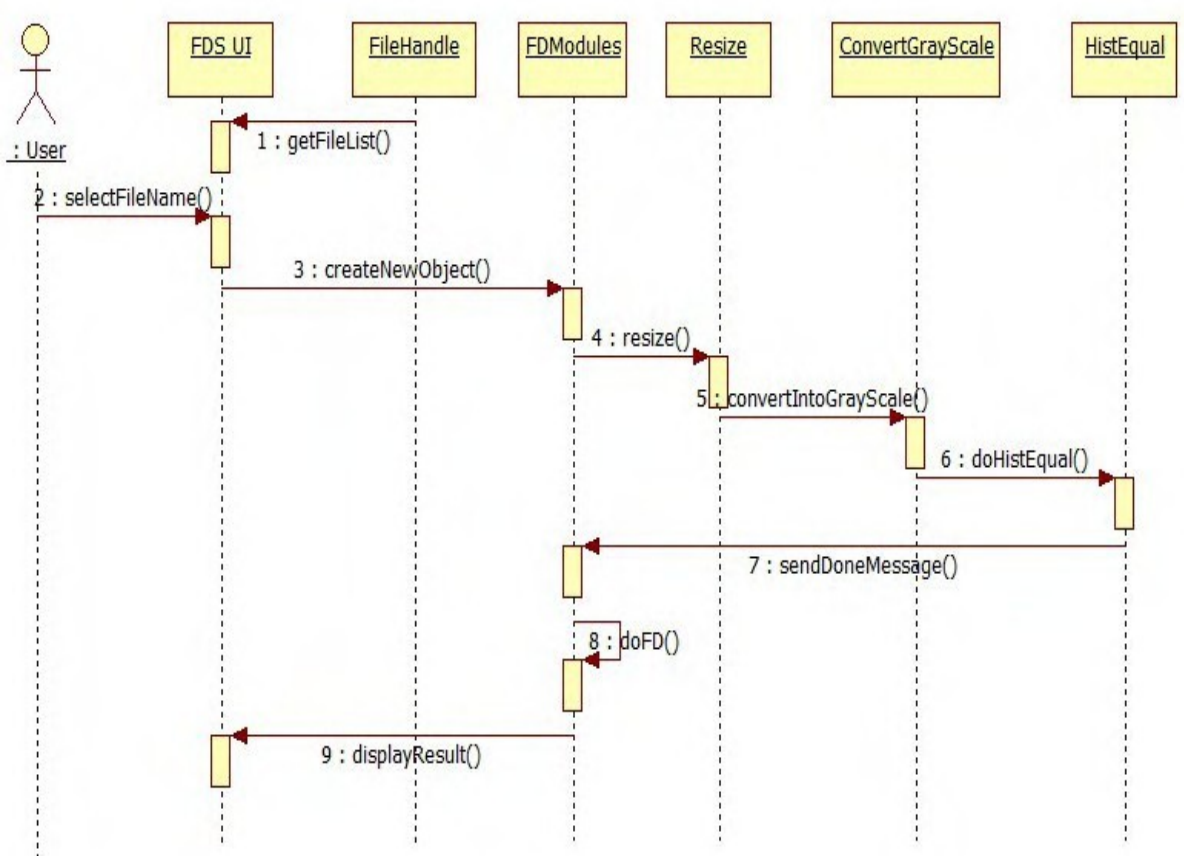


Figure 4.4 :sequence diagram of face detection

Sequence Diagram – Preparation Sample

The following diagram is sequence diagram to show how to prepare the sample.

1 Trainer select sample folder

2 Train Tools call “opencv_create sample” and execute itself.

3 output screen is shown and acknowledge trainer.

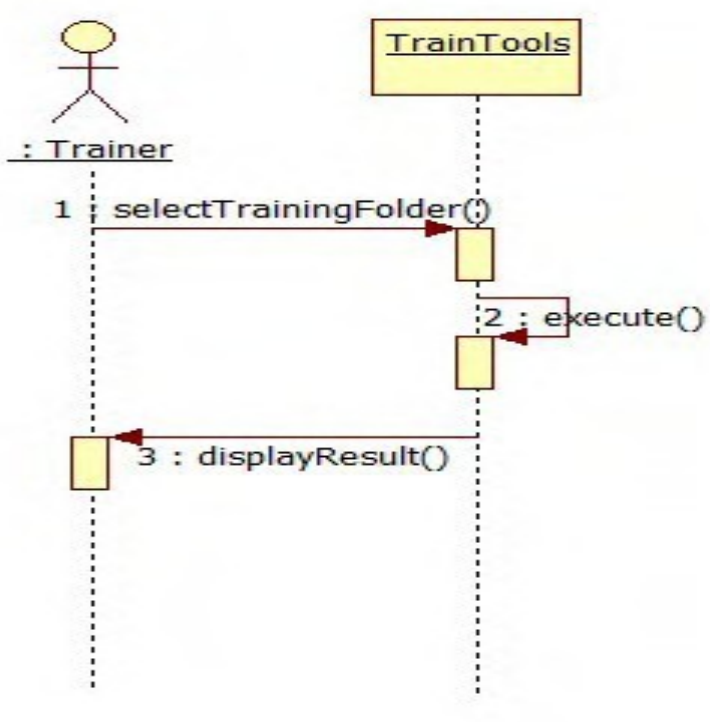


Fig 4.5 :Sequence Diagram – Preparation Sample

Sequence Diagram – Training Sample

After the step of preparation sample, the following diagram is illustrated how to train the sample step by step.

1 Trainer select vector file which is generated from before.

2 Trainer activate “opencv_haar training” utility to perform training the sample.

3 XML file is generated when training is finished.

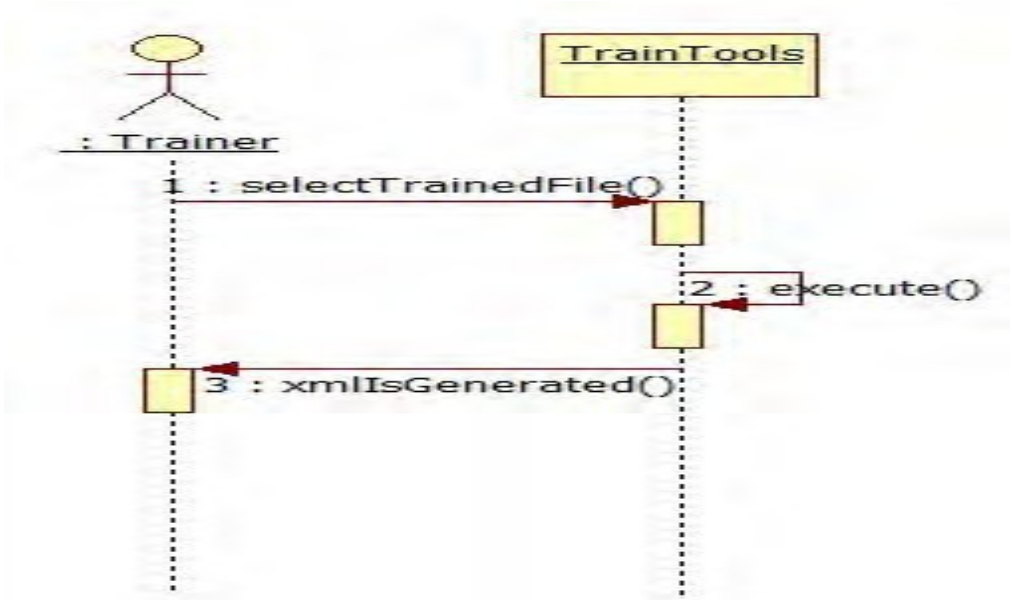


Fig 4.6 :Sequence Diagram – Training Sample

Activity Diagram

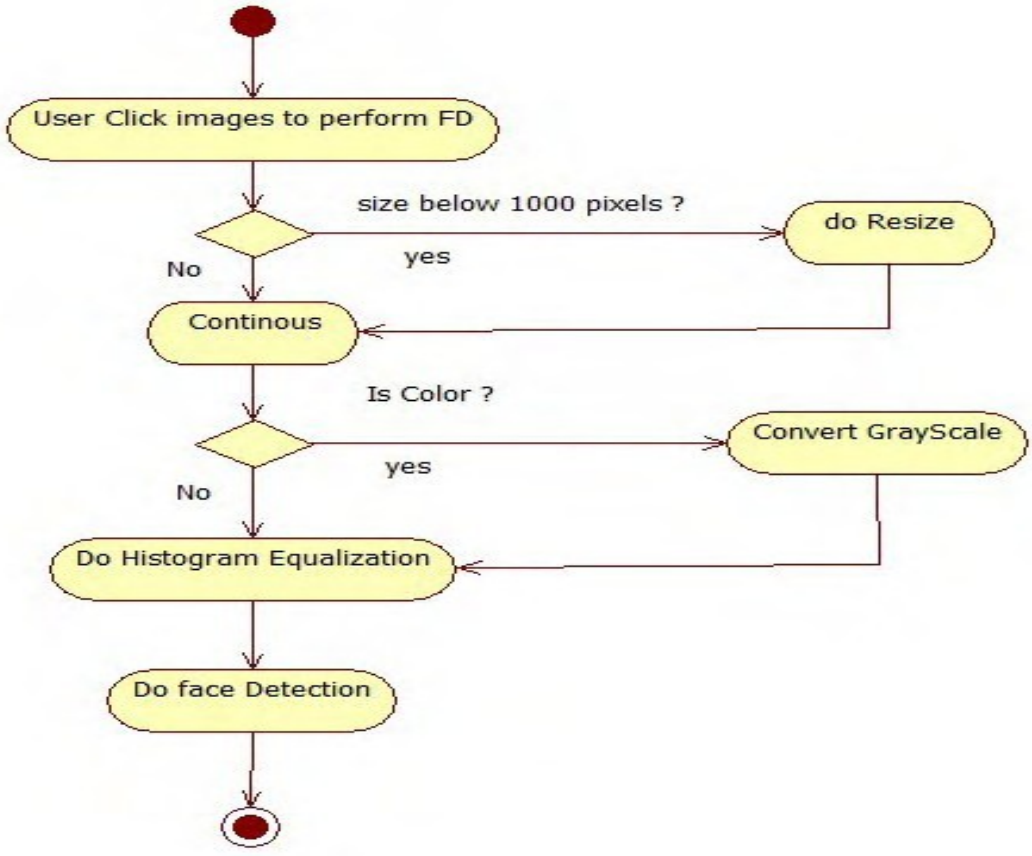


Fig 4.7 : Activity Diagram

The following diagram is workflows of face detection to show system's operation.

Class Diagram

The following figure is shown the main class of FDS. The "FDS UI" is the user interface class mainly handle any GUI related work. For example, update the UI widget. The "FileHandle" class is handled any file operation. For example, check the file is exists or not. The "ImageProcess" class is the image processing tools. For example, it handles resizing, convert into greyscale. The "FD Module" class perform face detection.

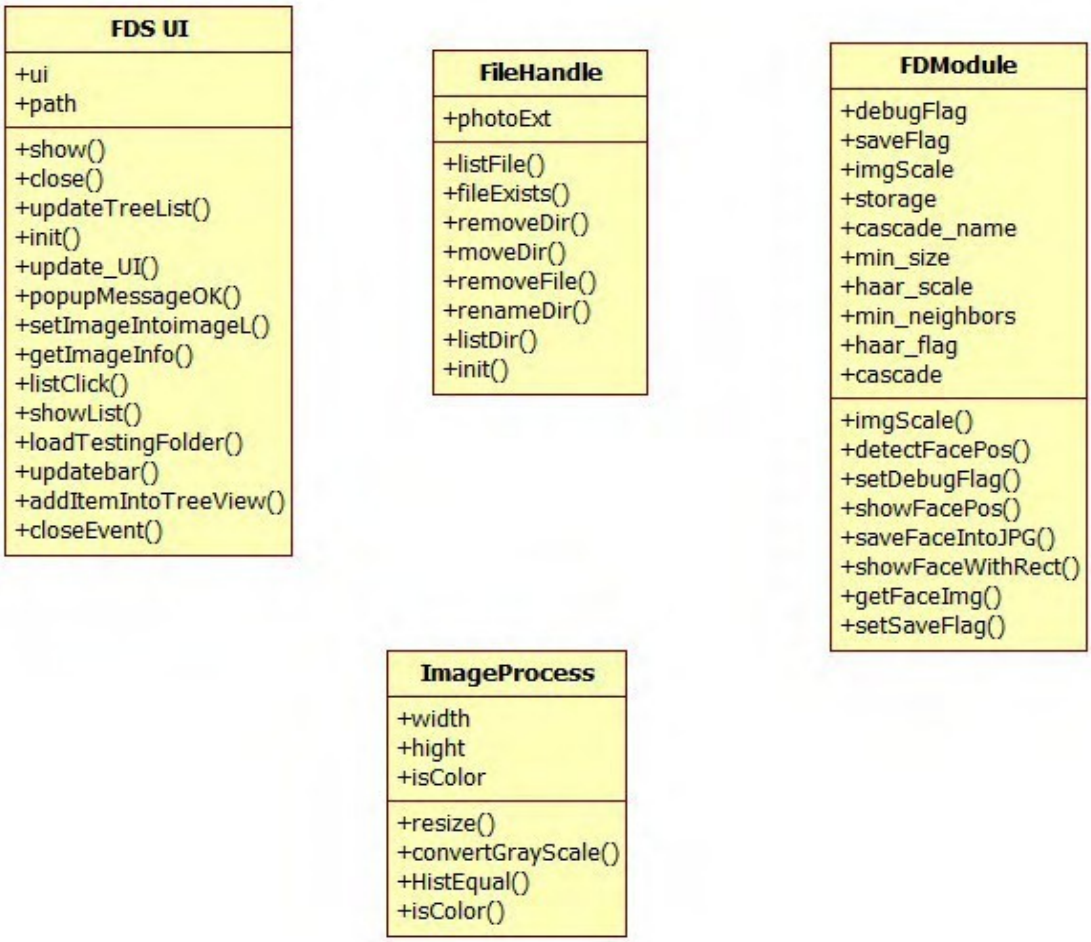


Fig 4.8 : Class diagram

4.4 Summary

This section presents UML diagram to give the ideal about the analysis and design approaches of the project.

CHAPTER 5
PROJECT IMPLEMENTATION PLAN

5.1 WORKING PLAN

Introduction

In this project, we observe the entire work structure, meaning how the scheduling was maintained throughout the developmental phase. We shall also see the financial foundation of this project and furthermore the feasibility study should be also discussed.

5.1.1 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-2	Study related works using OpenCV
Week-2	Study related works using Bluetooth
Week-3	Study related works using processing
Week-3	Study image processing
Week-3	Study image processing
Week-4	Sketching basic structure
Week-4	Prototype design
Week-4	Finalize Prototype design
Week-4	Flexible Box
Week-5	Runnable with basic commands(Input, Output,Turn on, Turn Off)
Week-5	Designing Look ahead table
Week-5	Designing Look ahead table
Week-6	Creating environment for image processing

Week-6	**Creating environment for image processing**
Week-7	**Integrating all together**
Week-7	**Start coding**
Week-8	**Coding for basic instructions (Compare,Result, Accuracy measure)**
Week-8	**Coding for single face detection**
Week-9	**Single face detection and Compare with database**
Week-9	**Multiple Face detection and Compare**
Week-10	**Detecting Multiple face, store and compare with database**
Week-10	**Attendance collection**
Week-10	**File Generate base on collective data**
Week-10	**Daily file generation of attendance**

Table 5.1 : Work Plan

5.1.2 Gantt Chart

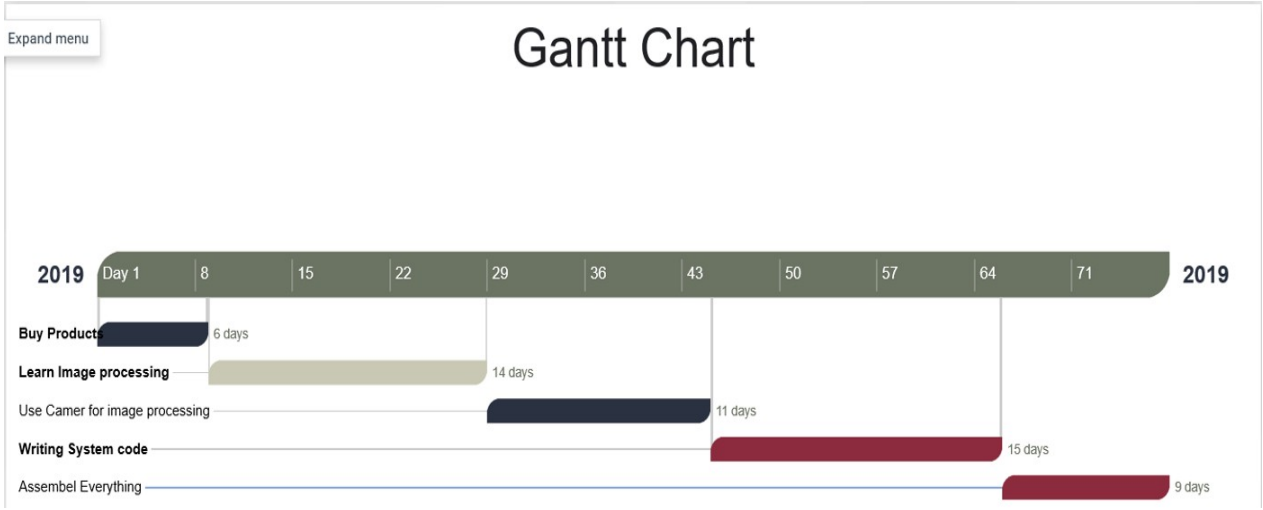


Fig 5.1 : Gantt Chart

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

- Form a project team and appoint a project leader.
- Enumerate potential proposed system.
- Define and identify characteristics of proposed system.
- Determine and evaluate performance and cost effectively of each proposed system.

- Weight system performance and cost data.
- Select the best-proposed system.
- Prepare and report final project directive to management.

5.1.3 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.

5.1.4 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.

5.1.5 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?

5.1.6 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.

5.1.3 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.

5.2 CODE IMPLEMENTATION

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

```
FRAS
|___Dataset
|___main.py
|___dataset.py
|___database.log
|___data_set.csv
|___data_log.ods
```

All those file in the project directory.

```
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.
```

5.2.1 main.py

All the work will be done here, Detect the face ,recognize the faces and take attendance.

```
import os  # accessing the os functions
import check_camera
import Capture_Image
import Train_Image
import Recognize
# creating the title bar function
def title_bar():
    os.system('cls')  # for windows
    # title of the program
    print("\t**********************************************")
    print("\t***** Face Recognition Attendance System *****")
    print("\t**********************************************")
# creating the user main menu function
def mainMenu():
    title_bar()
    print()
    print(10 * "*", "WELCOME MENU", 10 * "*")
```

```
    print("[1] Check Camera")
    print("[2] Capture Faces")
    print("[3] Train Images")
    print("[4] Recognize & Attendance")
    print("[5] Auto Mail")
    print("[6] Quit")
    while True:
        try:
            choice = int(input("Enter Choice: "))
            if choice == 1:
                checkCamera()
                break
            elif choice == 2:
                CaptureFaces()
                break
            elif choice == 3:
                Trainimages()
                break
            elif choice == 4:
                Recognize()
                break
            elif choice == 5:
                os.system("py automail.py")
                break
                mainMenu()
            elif choice == 6:
                print("Thank You")
                break
            else:
                print("Invalid Choice. Enter 1-4")
                mainMenu()
        except ValueError:
            print("Invalid Choice. Enter 1-4\n Try Again")
    exit
def checkCamera():
    check_camera.camer()
    key = input("Enter any key to return main menu")
```

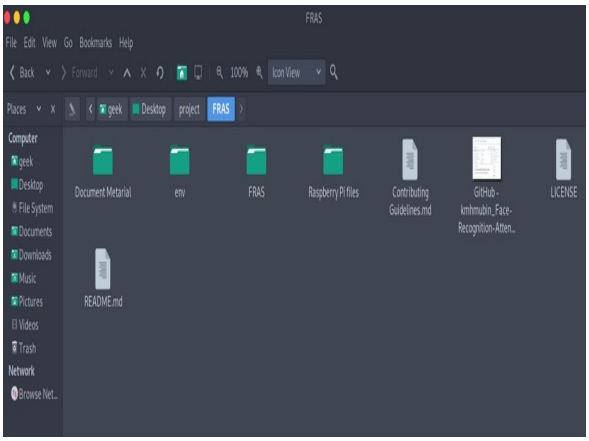
```
    mainMenu()
def CaptureFaces():
    Capture_Image.takeImages()
    key = input("Enter any key to return main menu")
    mainMenu()
def Trainimages():
    Train_Image.TrainImages()
    key = input("Enter any key to return main menu")
    mainMenu()
def RecognizeFaces():
    Recognize.recognize_attendence()
    key = input("Enter any key to return main menu")
    mainMenu()
MainMenu()
```

5.2.2 Dataset.py

Dataset Implementation code are given below which is also in python code.

```
1.1   import cv2
1.2   from picamera.array import PiRGBArray
1.3   from picamera import PiCamera
1.4   import time
1.5   import os
1.6   import numpy
1.7   import io
1.8   stream = io.BytesIO()
1.9   cam = cv2.VideoCapture(0)
1.10  detector=cv2.CascadeClassifier('haarcascade_frontalface_de fault.xml')
1.11  sampleNum=sampleNum+1
1.12  cv2.imshow('frame',img)
1.13  if cv2.waitKey(100) & 0xFF == ord('q'):
1.14  break
```

5.2.3 Sample Images



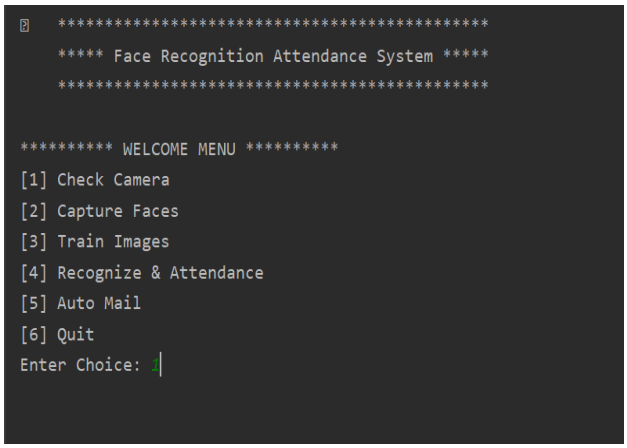


Fig 5.2.1 : FRAS User interface

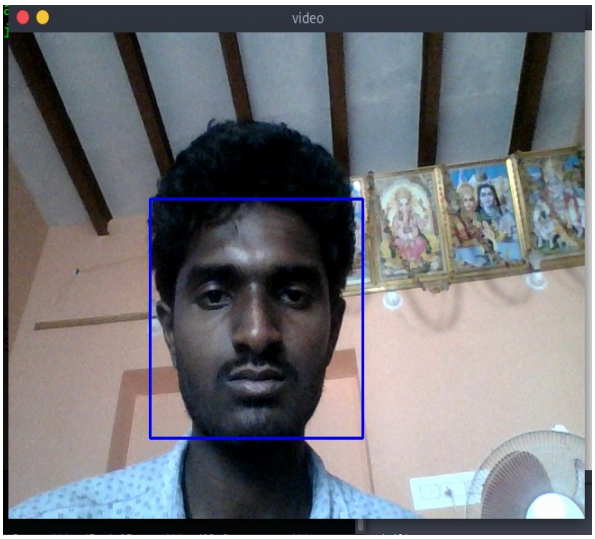
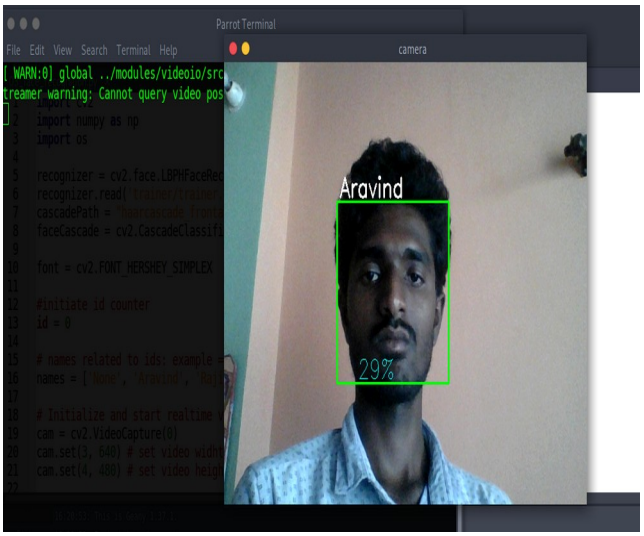


Fig 5.2.2 : Capturing Face



5.2.3 Recognised face

5.2.4 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 – 6

TESTING

6.1 Software Testing

Testing is the technique for assessing a framework or its parts to search out whether it fulfills the ideal requirements or not. in direct words, testing is executing a framework to recognize any holes, mistakes, or missing requirements. It is a fundamental advance in a product designing cycle to develop any product. Software Testing is fundamental since we as a whole commit errors. A portion of those errors are irrelevant, however, some of them are costly. We have to check everything and anything we produce since things can generally turn out wrong.

6.2 Black box testing

In black-box testing, the structure of the program isn't thought about. It considers the usefulness of the application as it was. It is likewise called practical testing. In this sort of testing, the analyzers focus on utilitarian testing, that is, on giving known information and checking if the realized yield is acquired. This strategy is by and large followed while completing acknowledgment testing when the end-client isn't a product engineer however just a client.

6.3 White-box testing

White-box Testing is a kind of testing technique where the tester (a developer testing the application) knows about the framework internals. Its usage is straightforward. The goal is to guarantee that each line of the code is tried. The tester recognizes all consistent, plan, and typographical mistakes. The tester likewise needs to approve the interior structure of the thing viable alongside the yield.

6.4 CASE Tools

Computer-Aided Software Engineering (CASE) is the computer computerization of framework advancement exercises. These can be arranged into 3 classes: front-end, back-end, or cross life cycle CASE items, as indicated by when they are utilized during frameworks improvement. Front-end apparatuses now and again called capitalized devices, are utilized during primer examination, venture arranging, investigation, or plan. Back-end instruments, here and there called lower-CASE devices, are utilized during advancement, execution, assessment, and upkeep. Cross life cycle CASE apparatuses are utilized to help progressing exercises, for example, venturing the board and making documentation, which happens over numerous periods of framework improvement.

CASE apparatuses can likewise be free or coordinated. Framework documentation is regularly done utilizing CASE instruments that are autonomous of one another, to computerize explicit

advancement exercises, for example, information stream outlines or ER diagrams. In any case, these individual apparatuses may not have similar organizations for sharing data.

Future upkeep of the framework at that point turns out to be incredibly troublesome. Interestingly, incorporated CASE climate centers around the upkeep of configuration records, and CASE devices are utilized to deal with exercises all through the whole framework advancement. Incorporated CASE devices utilize a typical vault with the goal that data can be shared over all instruments and framework improvement exercises. Any progressions to the plan records typically bring about the most different parts of the framework mirroring those changes.

6.5 Requirement Traceability Matrix

A traceability matrix is a document that associates any two-gauge records that require a many-to-numerous relationship to check the fulfillment of the relationship. It is utilized to follow the prerequisites and to check the current venture necessities are met.

Requirement Traceability Matrix, RTM catches all necessities proposed by the customer client and advancement group client and their detectability in a solitary archive conveyed after the life-cycle.In different words, it is a report that guides and follows client necessities with experiments. The fundamental reason for the Requirement Traceability Matrix is to see that all experiments are covered so no usefulness should miss while testing.

Functional Requirement - ID	Functional Requirements Description
FR-1	Face Detection
FR-2	Capturing Faces and Training Model
FR-3	Storing images in the database
FR-4	Face Recognition
FR-5	Recording Attendance
FR-6	Emailing Attendance

Table 6.1 : Functional Requirements Tables

Test Case - ID	Test Case Description
TC-1	Web camera detecting faces live
TC-2	Capturing multiple images
TC-3	Storing images with student details

TC-4	Encoding faces
TC-5	Training model with training images
TC-6	Recognizing faces registered in the database
TC-7	Marking attendance along with time
TC-8	Emailing the attendance along with details of timestamp

Table 6.2: Test Cases

6.6 Test Plan

The main idea behind the test plan consists of testing the individual parts separately on the first stage for a unit test. Then merge all modules step by step for a final test. Therefore all modules functionality is validated through a unit test and the system functionality is verified through a final test.

Test Cases and Outputs

1: Web camera detecting faces live

The software locates the eyes and mouth, we can also rotate, scale, and shear the image which results in proper centering of the eyes and mouth to its best.

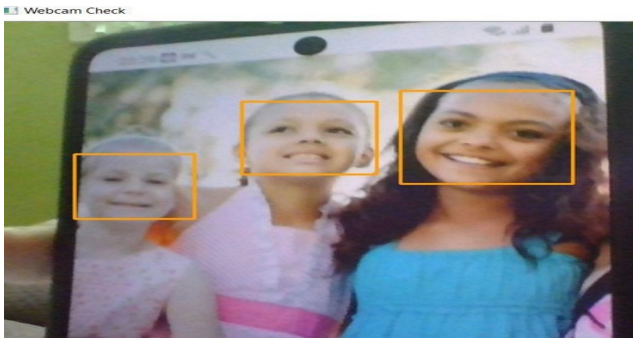


Fig 6.1. Detection of multiple faces

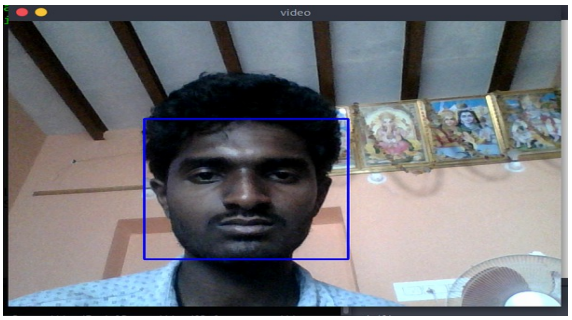
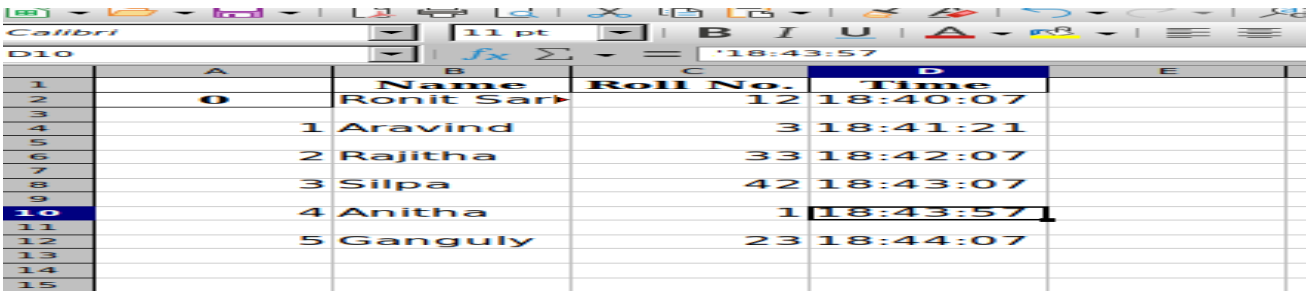


Fig 6.2. Face detection

3: Storing images with student details

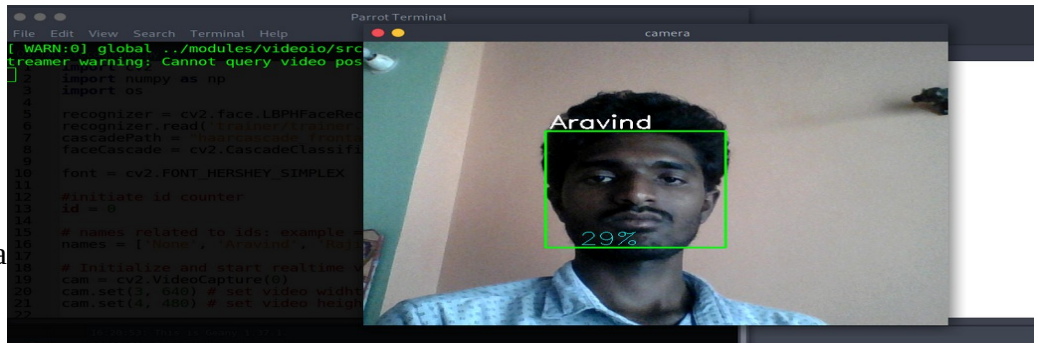


	A	B	C	D	E
1		Name	Roll No.	Time	
2	0	Ronit Sar	12	18:40:07	
3					
4	1	Aravind	3	18:41:21	
5					
6	2	Rajitha	33	18:42:07	
7					
8	3	Silpa	42	18:43:07	
9					
10	4	Anitha	1	18:43:57	
11					
12	5	Ganguly	23	18:44:07	
13					
14					
15					

Fig 6.3 :Storing Data

4: Training Model with captured images

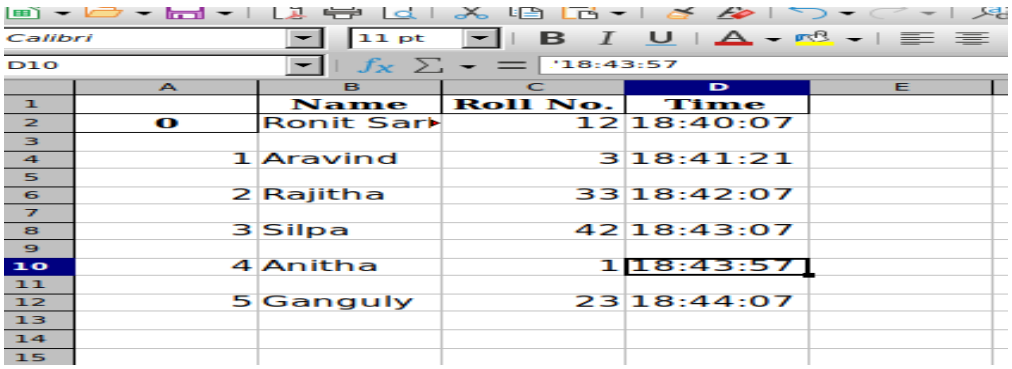
5.Recognizing faces registered in the database



6:Ma

rkin

g Attendance



	A	B	C	D	E
1		Name	Roll No.	Time	
2	0	Ronit Sar	12	18:40:07	
3					
4	1	Aravind	3	18:41:21	
5					
6	2	Rajitha	33	18:42:07	
7					
8	3	Silpa	42	18:43:07	
9					
10	4	Anitha	1	18:43:57	
11					
12	5	Ganguly	23	18:44:07	
13					
14					
15					

6.7 Summary

Test Report is a document which contains a summary of all test activities and final test results of a testing project. Test report is an assessment of how well the Testing is performed. Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software release

CHAPTER 7
HANDIWORK

7.1 Introduction

This project output 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.

Results and Conclusions

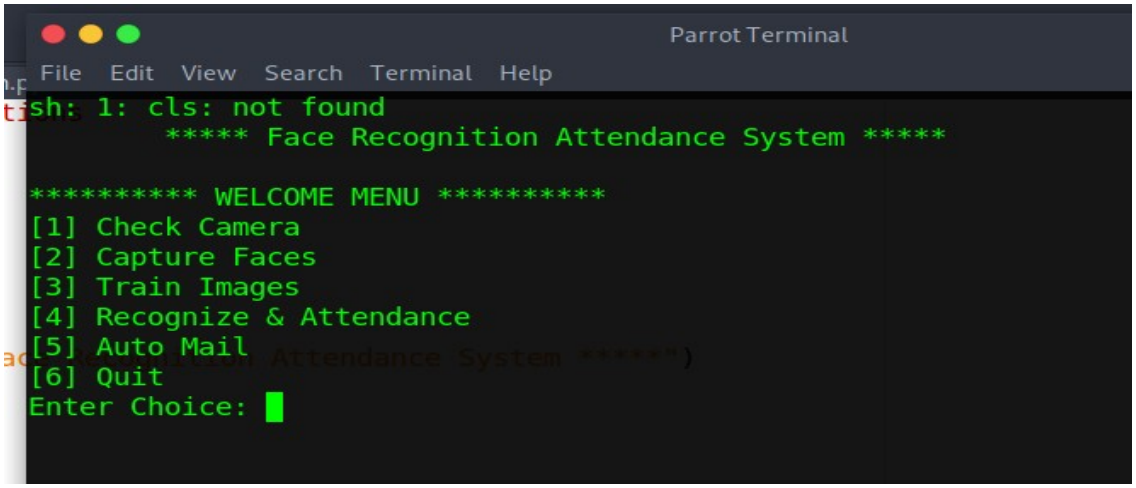


Fig 7.1: User Interface

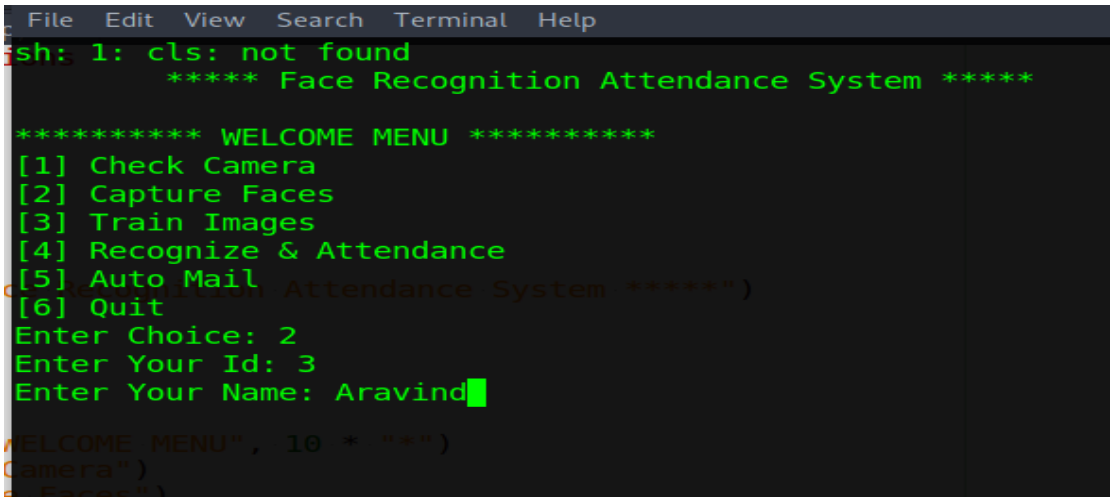


Fig 7.2 : Interacting with UI

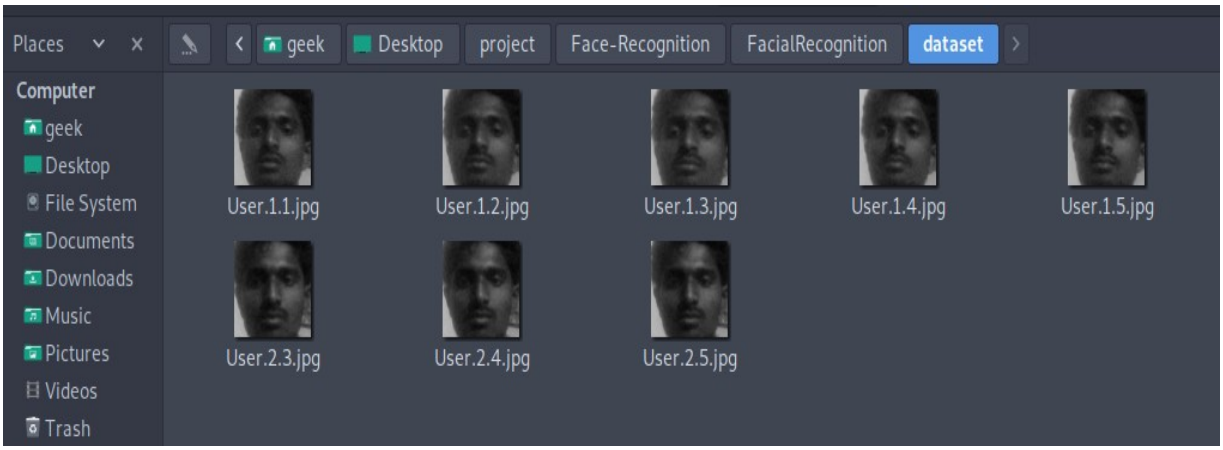


Fig 7.3 : Image Dataset

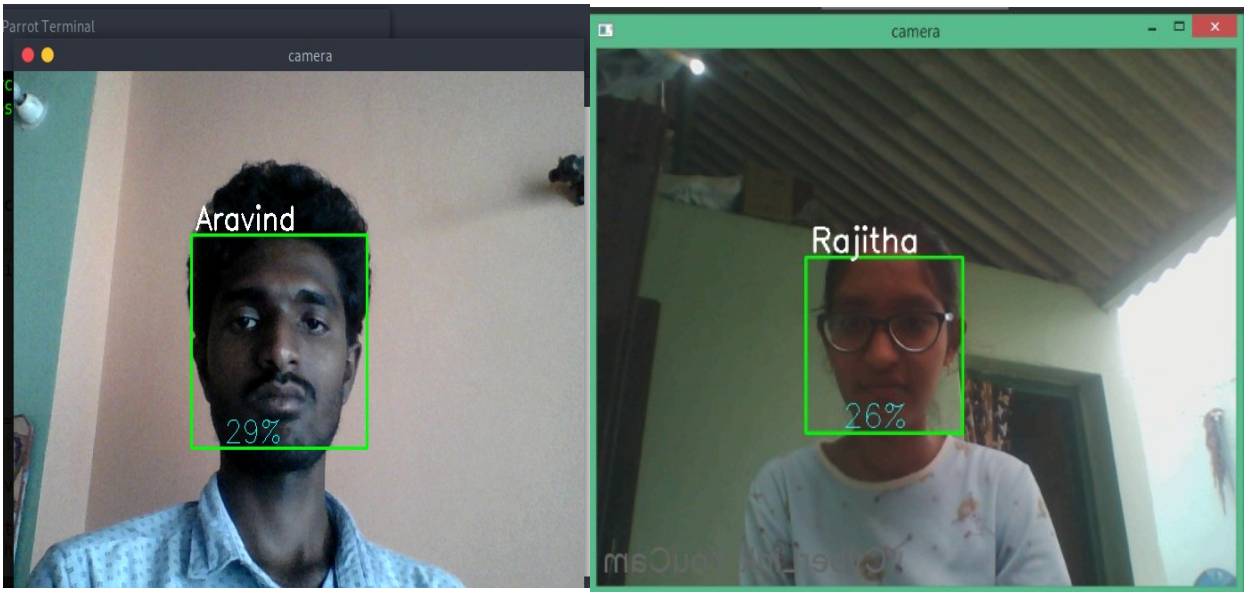
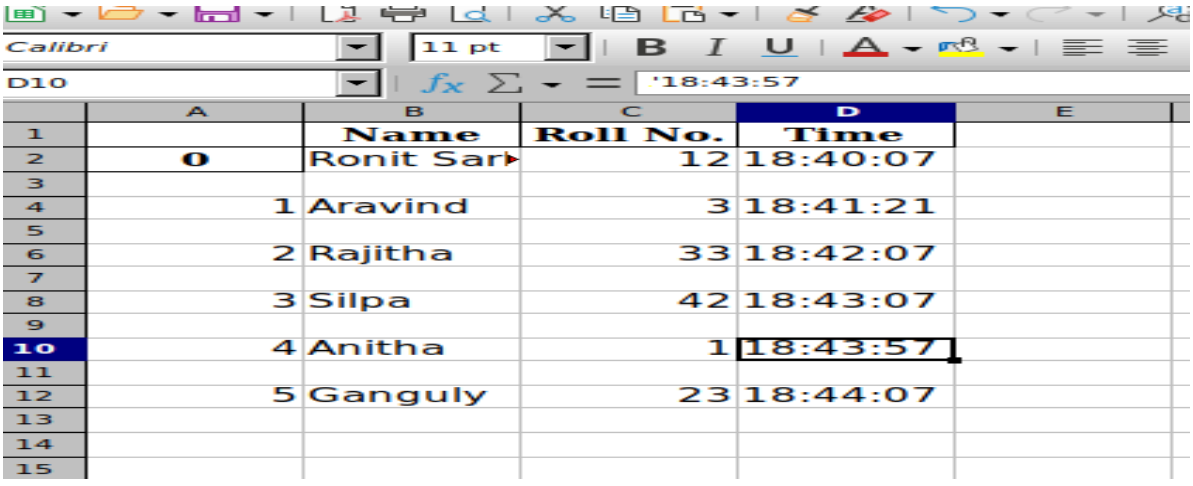


Fig 7.4 Recognizing Faces



	A	B	C	D	E
1		Name	Roll No.	Time	
2	0	Ronit Sar	12	18:40:07	
3					
4	1	Aravind	3	18:41:21	
5					
6	2	Rajitha	33	18:42:07	
7					
8	3	Silpa	42	18:43:07	
9					
10	4	Anitha	1	18:43:57	
11					
12	5	Ganguly	23	18:44:07	
13					
14					
15					

Fig 7.5 :Marking Attendace

While testing the working model, we chose some pictures of ourselves, our friends, and some well-known celebrities. It turned out that the hit percent was pretty satisfying, i.e around 85%. Below is the observation table containing the information about the number of times faces got recognized correctly. The test was performed around 40-50 times and it includes multiple faces in the same image as well.

Sr. No.	ID of person	No. of pictures	No. of times tested	No. of hits
1.	**14**	35	10	9
2.	**1**	40	10	9
3.	**20**	35	10	8
4.	**17**	37	10	7
5.	**6**	30	15	15
6.	**30**	35	15	14
7.	**33**	40	15	13

Table 7.1: Face recognition results and hit rates

The face detection shows 88.23% of the right hit rate.

An important step is filtering out non-facial test images from an image. Several approaches have been taken, but it was not easy to find an absolute threshold value which can be applied to various pictures with different light conditions and composition. Approaches using luminance, average brightness, etc., have been tried, but they turned out not to be good enough to set an appropriate threshold for filtering out non-facial test images. Lastly, a statistical method was tried.

After filtering out the leftmost column elements, which are12 test images. Out of 21 faces in the picture, the algorithm has detected 19 faces within the acceptable error of the location of faces. The two undetected faces are partially blocked by the other faces. Conclusively, this statistical approach, which is a seemingly rough estimation, works great in this picture and it turned out to produce good results for other pictures as well.

7.2 Summary

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

CHAPTER – 8

FUTURE WORK & CONCLUSION

8.1 Introduction

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

8.2 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.

8.3 Conclusion

Smart attendance management system is designed to solve the issues of existing manual systems. We have used face recognition concept to mark the attendance of student and make the system better. The system performs satisfactory in different poses and variations.

In future this system need be improved because these system sometimes fails to recognize students from some distance, also we have some processing limitation, working with a system of high processing may result even better performance of this system.

In this project, FDS is constructed and achieve highly detection rate at least over 80% for Test Case 004 (CMU test set), which faces in images are uncontrolled environment.

In the fontal face Images testing, FDS got 95.99% of detection rate from Test Case 002 (BioID), testers were all in indoor condition. That means FDS works very well with indoor activities or under controlled environment.

In the variation of pose testing, FDS got 88.15% of detection rate from Test Case 003. It means FDS can be handled controlled pose of human face.

8.4 Limitations

Although, FDS performs highly detection rate, achieved 80% detection rate for all test case, in this project. However, all the miss detected face can be concluded as follows:

1.illumination condition,

2.pose orientation,

3.facial expression and

4.conclusion

In **Test Case 002**, 61 images fail to hit as a face, it is because faces are blocked (occlusion). In other words, faces aren't shown fully, and face detector classified as a non-face object. There has 61 images are regarded as non-face objects. The following image is the example of regarding as non-face objects.

In **Test Case 003**, most of the faces are missed and reported as non-face images, it is because most of the face has unacceptable variation of pose. The following image is the example of "variation pose".

In conclusion, this project presents a face detection system (FDS) using AdaBoost (machine learning algorithms). It got at least 80% of face detection rate. This method is suitable for the binary classification problem, whatever in any environment.

8.5 Summary

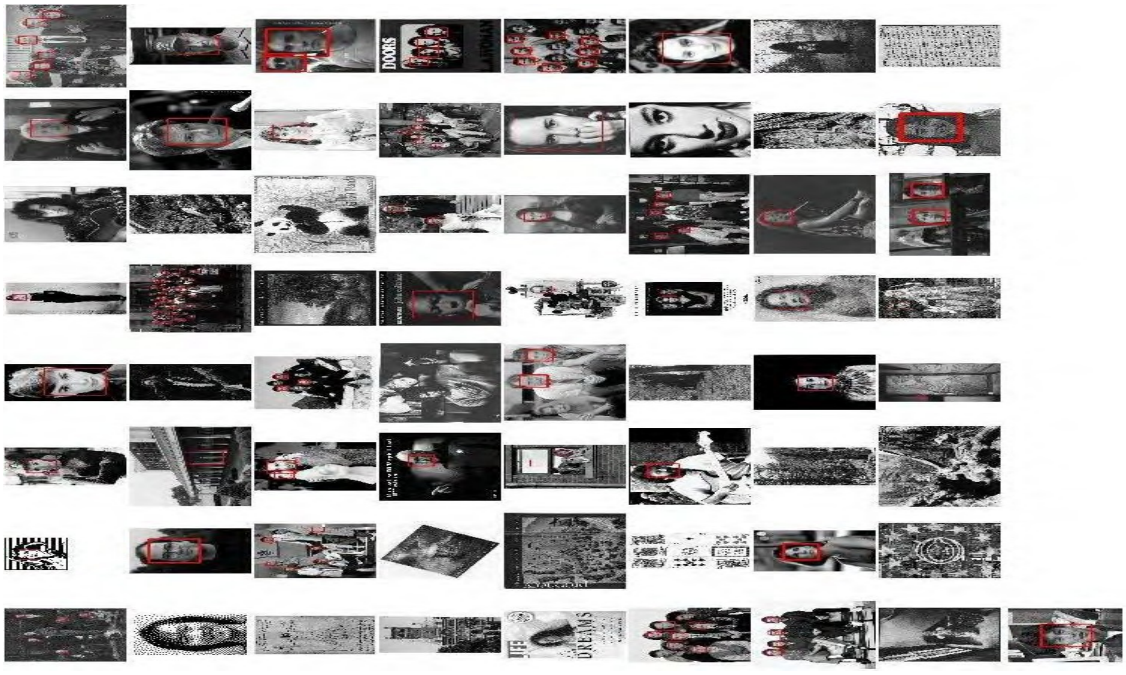
This project has described the possible future applications of the design. But there are a lot of possibilities with the designed device. The device may need some research for different applications, though the principle of the designed system will remain as it is.

Appendix

Result of Test case Sample 001

Result of Test case Sample 002

- **Result of Tesat case Sample 003**



BIBLIOGRAPHY

1.1 Computer vision - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Computer_vision accessed 14 June 2009

1.2 BBC NEWS | UK | Britain is 'surveillance society', http://news.bbc.co.uk/2/hi/uk_news/6108496.stm accessed on 14 June 2009

1.3 Python (programming language) - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Python_%28programming_language%29 access May 2009

1.4 Guido van Rossum - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Guido_van_Rossum, access May 2009

1.5 PyQt4 - PythonInfo Wiki, http://wiki.python.org/moin/PyQt4 access June 2009

1.6 Riverbank | Software | PyQt | What is PyQt?, http://www.riverbankcomputing.co.uk/software/pyqt/intro, access June 2009

1.7 SourceForge.net: Open Computer Vision Library, http://sourceforge.net/projects/opencvlibrary/, accessed May 2009

1.8 OpenCV – Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/OpenCV, accessed May 2009

1.9 Welcome – OpenCV Wiki, http://opencv.willowgarage.com/wiki/, accessed May 2009

1.10 DARPA Grand Challenge – Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/DARPA_Grand_Challenge, accessed March 2009

1.11 Face detection – Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Face_detection, accessed December 2008

1.12 Face detection - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Face_detection accessed 14 June 2009

1.13 Yang W.Q., B.L. Goulart, K. Demchak, and Y.D. Li. 2002. Interactive effects of ycorrhizal inoculation and organic soil amendments on nitrogen acquisition and growth of highbush blueberry. Journal of the American Society for Horticultural Science. Vol. 127, No. 5. p. 742-748.

1.14 An Introduction to Face Recognition Technology, http://inform.nu/Articles/Vol3/v3n1p01-07.pdf accessed May 2009

1.15 Human–computer interaction - Wikipedia, the free encyclopedia,