FACE RECOGNITION USING “HAAR CASCADE” TECHNIQUE FOR DOOR UNLOCKING SYSTEM

A Project report submitted in fulfilment of the requiremets

for the awards of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

M Vasavi(316126512033)	K Raja (316126512054)
P Avinash (316126512044)	Y Sai Mahidhar(316126512059)

Under The Guidance Of

Mr.B Chandra Mouli,M.Tech

Assistant professor , Department of ECE

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY & SCIENCES

(Affiliated to Andhra University, Approved by AICTE &Accredited by NBA)

SANGIVALASA, BHEEMILI MANDAL, VISAKHAPATNAM DIST.(A.P)

(2019-2020)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY & SCIENCES

(Affiliated to Andhra University, Approved by AICTE & Accredited by NBA)

Sangivalasa, Visakhapatnam Dist.,(A.P.)

CERTIFICATE

This is to certify that this project report entitled **"Face Recognition Using "Haar Cascade" Technique For Door Unlocking System"** has been submitted by **M Vasavi (316126512033), K Raja (316126512054), P Avinash (316126512044), Y Sai Mahidhar (316126512059)** in fulfilment of the requirements for the award of **"Bachelor of Technology"** in **Electronics Communication Engineering**.

HEAD OF THE DEPARTMENT

Dr.V.Rajya Lakshmi
M.E., Ph.D., MIEEE, AMIETE, AMIE
Professor
Department of ECE
ANITS

PROJECT GUIDE

B Chandra

Mr.B Chandra Mouli
M.Tech
Assistant Professor
Department of ECE
ANITS

ACKNOWLEDGEMENT

We express our heartful thanks to **Mr.B Chandra Mouli, Assistant Professor**, Department of Electronic and Communication Engineering. Anil Neerukonda institute of Technology and Sciences (ANITS), for his generous help, valuable guidance, and cooperation in completing our project successfully.

We express our deep sense of gratitude and respect to **Dr.V.Rajya Lakshmi,** Professor & Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work. We express our thanks to all teaching and non-teaching staff of Department of ECE, for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

M Vasavi(316126512033)

K Raja(316126512054)

Avinash(316126512044)

Y Sai Mahidhar(316126512159)

CONTENTS

Pg-No

CERTIFICATE 02

ACKNOWLEDGEMENT 03

LIST OF FIGURES 06

ABSTRACT 08

CHAPTER 1 INTRODUCTION 09

1.1 Project Objective 09

1.2 Project Outline 09

1.3 Disadvantages of other technologies for security systems 10

CHAPTER 2 INTRODUCTION TO FACE RECOGNITION 11

2.1 Introduction 11

2.2 Background and related work 11

2.3 Feature extration 12

2.4 Face matching 12

2.5 Techniques 12

2.6 Algorithms 14

2.7 Applications of Face Recognition 14

CHAPTER 3 IMAGE PROCESSING 16

3.1 Introduction 16

3.2 Purpose of image processing 16

3.3 Types of image processing 16

3.3.1 Analog image processing 17

3.3.2 Digital image processing 17

3.4 Image Processing Techniques 18

3.4.1 Image re-sizing 19

3.4.2 Image filtering 19

3.4.3 Image segmentation 20

3.4.4 Edge detection 20

3.4.5 Image enhancement 21

3.5 Applications of Image processing 22

CHAPTER 4 FACE DETECTION USING HAAR CASCADE METHOD 23

4.1 Introduction 23

4.2 Haar Cascades 24

4.3 Feature Extraction 24

4.4 Haar Features Applied to Image 26

4.5 Adaboost 27

4.6 Cascading 27

CHAPTER 5 HARDWARE COMPONENTS 30

5.1 Raspberry Pi 30

5.2 Pi Camera 32

5.3 Raspberry Pi OLED display 34

CHAPTER 6 OPEN CV 36

6.1 Introduction 36

6.2 History 36

6.2.1 Computer Vision 37

6.3 Testing Open CV 40

6.4 Applications of Computer Vision 42

6.4.1 Event Detection 43

6.4.2 Medical Image Processing 44

6.4.3 Self Driving Vehicles 44

6.4.4 Military Applications 44

CHAPTER 7 METHODOLOGY 45

7.1 Introduction 45

7.2 Block Diagram 49

7.3 Flow Chart 49

CHAPTER 8 RESULTS AND DISSCUSSIONS 54

CONCLUSIONS 56

REFERENCES 57

LIST OF FIGURES

Figure No.	Figure Name	Pg.No
2.1	Face recognition using geometric approach	13
2.2	Face recognition under different lightening conditions	14
3.1	various image processing techniques	18
3.2	Image filtering	19
3.3	Image segmentation	20
3.4	Edge detection	21
3.5	Image Enhancement	22
4.1	Haar feature extraction kernels	24
4.2	Face detection Kernels	25
4.3	kernels	25
4.4	Detection of in difference parts	26
4.5	Cascading	28
5.1	Raspberry pi	31
5.2	Kernel structure of raspberry pi	32
5.3	Raspberry pi and pi camera	33
5.4	Raspberry pi with OLED display	34
5.5	Raspberry pi with OLED circuit	35
7.1	Rough circuit diagram	46
7.2	Data Acquisition	47
7.3	Training	48
7.4	Recognition	48
7.5	Block diagram	49
7.6	Flow chart of methodology	49
7.7	installing dlib	51
7.8	install pillow	52

7.9 pip install face recognition -no-cache-dir 53

8.1 user identification 54

8.2 Random finger number generation 55

8.3 Random finger number identification 55

ABSTRACT

Conventional method of identification based on the exclusive knowledge like a social security number or a password are not all together reliable. Some how password can be forgotten or compromised. But a face is undeniably connected to its owner. It cannot be borrowed or stolen. Face recognition technology may solve this problem since a face is undeniably connected to its owner expect in case of identical twins. The system can compare scans to records stored in a central or local database. The software first captures an image of all the authorized persons and stores the information into the database.

Having many applications this project is mainly for future features like security, safety, monitoring and control to home automation. A day to day home security level grown up to provide security to our house based on face recognition can be implemented. A standard web camera to capture the image to identify the visitor. It's a method that identifies the visitor. If the face recognizes visitor, the door will be unlocked or opened. If they are not identified door will not be unlocked. The system will perform detection and recognition rapidly in real time when face in front of camera. This project basic utilizes the camera, and raspberry pi system to create a door unlocking mechanism. If the user at the door is recognized, door will be unlocked.

CHAPTER 1

INTRODUCTION

1.1 Project Objective:

In present day situation home security is big concern.The conventional system of lock and keys is obsolete and has limitations.Hence a door unlocking using face recognition technology is more secure and efficient.

1.2 Project Outline:

This work presents a design and framework for a unlocking of a door or a any security device mechanism using face recognition which tackles the problem of security.Fingerprint can be forged and password can be forgotten or hacked but a face is undeniably connected to the owner of face.

This system uses Haar Cascade technique along with Ada boot Algorithm for face detection, with Raspberry Pi as its core processing unit while using a 5 Mega Pixel camera to capture the face of the user.The system compares the face of the user with a pre defined database of authorized owners and opens the door if the face matches with the image in the database.The face is scanned and compared in three different angles for maximum security and is equipped with a motion sensor which acts as a foolproof against images,pictures etc.

1.3 Disadvantages of other technologies for security systems:

1. RFID system
2. Biometric finger print system
3. Password or Pin

1.3.1 Disadvantages of RFID:

1. If a barcode is ripped or damaged there is no way to scan the product.
2. RFID still has two separate chips (read only and readable/writable),which cannot be read by the same machine.

1.3.2 Disadvantages of finger printing system:

1. It can make mistakes with the dryness or dirty of the finger’s skin,as well as with the age(is not appropriate with children,because the size of their fingerprint changes quickly).
2. Slow due to one by one process.

1.3.3 Disadvantages of Pin or Password:

1. Can be forgotten.
2. It can be hacked and be seen by a person while the administrator enters the pin or password.

CHAPTER 2

INTRODUCTION TO FACE RECOGNITION

2.1 INTRODUCTION

Face recognition is a pattern recognition task performed on it on be described as classifying a face either "known" or “unknown”, after comparing it with stored individuals, It is also desirable to have system that has the Ability to recognize unknown faces.

2.2 BACKGROUND AND RELATED WORK

Much of the work in computer recognition of faces has focused on detecting individual features such as the eyes, mouth, and head outline, and defining the face model by the position, size, and relationships among these features

Face recognition has become an important issue in many applications such as security systems, credit card verification and criminal identification. For example. the ability to model a particular face and distinguish it from a large number of stored face models would make it possible to vastly improve criminal identification Even the ability to merely detect faces ,as opposed to recognize them, can be important Detecting faces in photographs for automating colour film development can be very useful, since the effect of many enhancement and noise reduction techniques depend on the image content.

Although there are three different approaches to the face recognition problems there are two basic methods from which these three different approaches arise. the first method is based on the information theory concepts, in other words, on the principal component analysis methods. In this approach ,the most relevant information that Fest describes a face is derived from the entire face image, Based on the K arhunen-Loeve expansion in pattern recognition, M.Kirbyand L.Sirovich have shown that any particular face could be economically represented in terms of a best coordinate system that they termed "Eigen faces". These are the Eigen functions of the averaged covariance of the ensemble of faces. Later, M.Turk and A.Pentland have proposed a face recognition method based on the Eigen Faces approach.

The second method is based on extracting feature vectors from the basic parts of a face such as eyes, nose, mouth, and chin. ln this method, with the help of deformable templates and extensive mathematics , key information from the basic parts of a face is gathered and then converted into a feature vector .

Face recognition is done in 2 steps:

I. Feature extraction

2. Face matching

2.3 FEATURE EXTRACTION

After the previous two steps, feature extraction is performed resulting in effective information that is useful for distinguishing between faces of different persons and stable with respect to the geometrical and photometrical variations.

2.4 FACE MATCHING

The extracted features are compared to those stored in the database, and decisions are made according to the sufficient confidence in the match score.

2.5 TECHNIQUES

Some facial recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw.These features are then used to search for other images with matching features.

Recognition algorithms can be divided into two main approaches:

a. Geometric approach

This is the historical way to recognize people, geometric features may be generated by segments, perimeters, and areas of some figures formed by the points. The featured set is studied to compare the recognition result. Distances in the feature space from a template image to every image in the database were calculated. Following to the FERET protocol,5 nearest face images were derived and if there were photos of the query person then the result was considered positive.

Each image was tested as a query and compared with others. The approach was robust, but its main problem is automatic point location . Some problem arises if image is of bad quality or several points are covered by hair

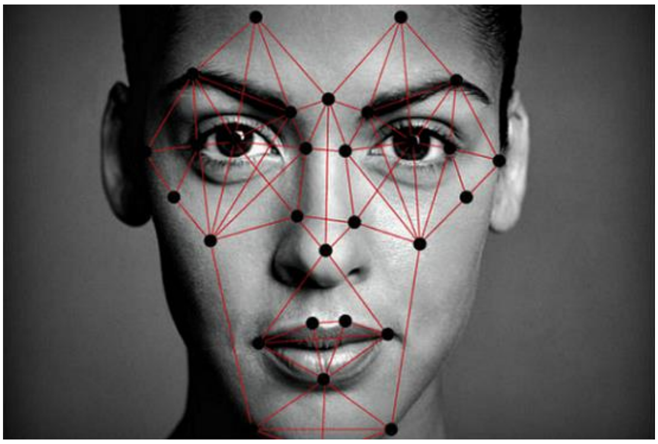


Fig 2.1 : face recognition using geometric approach

b. Photometric approach

It is a statistical approach that distils an image into values and compares the values with templates to eliminate variances. It relies on the image in the presence of light and the geometric location of different angles. The photometric transformation is implemented on the source image, does not take into account photometric changes, i.e., changes in the pixel. The main restriction in this approach is that multiple registered images of the same person are required .

Since it recognizes the new image by checking that it is spanned in a linear subspace of the multiple gallery images, it cannot handle the new images of a different person which is not included in the gallery set.

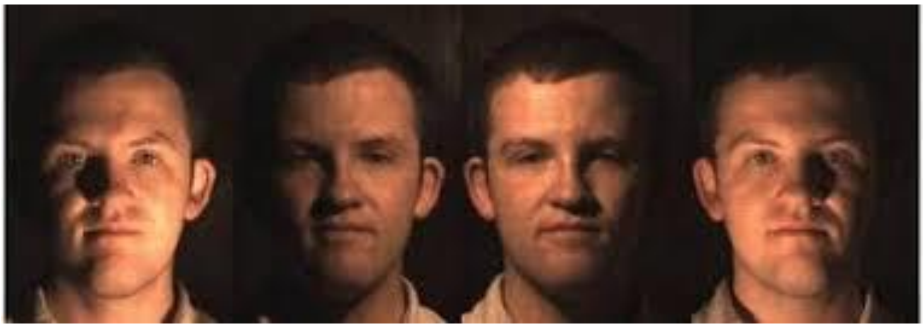


Fig2.2: Face images under different lightening conditions

2.6 ALGORITHMS

Popular recognition algorithms include Principal Component Analysis Using Eigenfaces, Linear Discriminate Analysis, Elastic Bunch Graph Matching Using the Fisher face algorithm, The Hidden Markov model, The Multilinear Subspace Learning using tensor representation and the neuronal motivated dynamic link matching.

2.7 APPLICATIONS OF FACE RECOGNITION

l. Because of certain limitations of fingerprint recognition systems, nowadays facial recognition systems are finding market penetration as Attendance monitoring alternatives.

2. Facial recognition systems are also beginning to be incorporated into unlocking mobile devices. The android market is working with facial recognition and integrating it into their cell phones. They have created an application called 'Visidon Applock'. This application allows you to put a facial recognition lock on any of your applications. This allows you to increase the safety of your private apps and content.

3. Also, in addition to biometric usages, modern digital cameras often incorporate a facial detection system that allows the camera to focus and measure exposure on the face of the subject, thus guaranteeing a focused portrait of the person being photographed. Some cameras, in addition, incorporate a smile shutter, or take automatically a second picture if someone closed their eyes during exposure.

4. In the 2000 presidential election, the Mexican government employed facial recognition software to prevent voter fraud. Some individuals had been registering to vote under several different names, in an attempt to place multiplevotes. By comparing new facial images to those already in the voter database, authorities were able to reduce duplicate registrations .Similar technologies are being used in the united states to prevent people from obtaining fake identification cards and driver's licenses.

CHAPTER 3

IMAGE PROCESSING

3.1 INTRODUCTION

In image science, Image Processing in any form of signal processing for which the input is an image, such as photograph or video frame, the output of image processing may be either an image or set of characteristics or parameters related to image. Image processing usually refers to digital image processing, but optical and analog image processing are also possible.

In simple terms, Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two dimensional signals while applying already set signal processing methods to them.

Image Processing basically includes the following three steps:

1.Importing the Image with optical scanner or by Digital photography

2.Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs

3.Output is the last stage in which result can be altered image or report that is based on image analysis.

3.2PURPOSE OF IMAGE PROCESSING

The purpose of image processing is divided into 4 groups they are:

1.visualization - Observe the objects that are not visible.

2.Image sharpening and restoration - to create a better image.

3.Image retrieval — Seek for the image of interest.

4. Measurement of pattern Measures various objects in an image

3.3 TYPES OF IMAGE PROCESSING

The two types of Image Processing used are:

I. Analog image processing

2. Digital image processing

3.3.1 Analog Image Processing

Analog image processing task conducted on two-dimensional analog signals by analog means. Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Association is another important tool in image processing through visual techniques. So analysts apply a combination of personal knowledge and collateral data to image processing.

3.3.2 Digital Image Processing

Digital Processing techniques helps in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre-processing, Enhancement and display, information extraction

In this case, digital computers are used to process the image. The image will be converted into the digital form using a scanner-digitizer and then process it. It is defined as the subjecting numerical representation of the objects to a series of operations in order to obtain the desired result. It starts with one image and produce a modified version of the image. It is therefore an image that takes one image into another.

3.4 IMAGE PROCESSING TECHNIQUES:

Digital image processing deals with manipulation and analysis of images by using computer algorithm, so as to improve pictorial information for better understanding and clarity. Image Processing involves the manipulation of images to extract information to emphasize or de-emphasize certain aspects of information, contained in the image or perform image analysis to extract hidden information. The computer vision system aims at recognizing objects if interest from given images and helps in developing the machine, that can perform visual function parallel to human vision. Computer vision system consists of Filtering, Coding, Enhancement, Restoration, feature extraction, analysis and recognition of objects from image. Processing of an image comprises of improvement in its appearance and effective representation of input image suitable for required application.

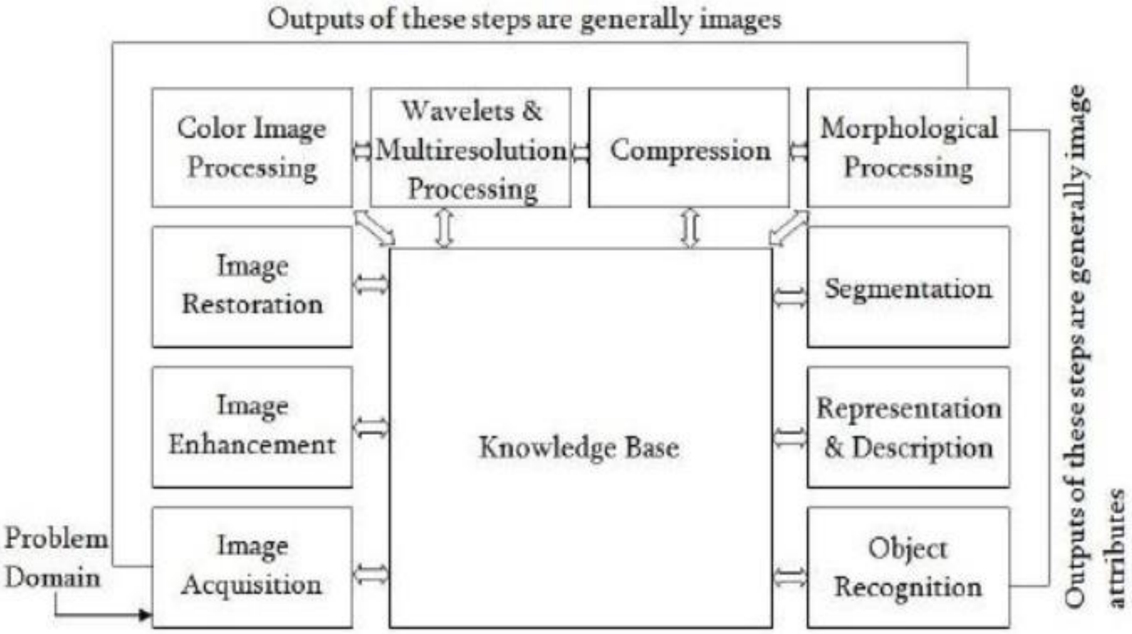


Fig 3.1: Various image processing techniques

3.4.1 Image re-sizing

Re-sizing of an image is performed by the process of interpolation. It is a process which re-samples the image to determine the values between defined pixels. Thus, resized image contains more or less pixels than that of original image. The intensity values of additional pixels are obtained through interpolation if the resolution of the image is increased.

3.4.2 Image Filtering

Uncertainities are introduced into the image such as random image noise, partial volume effects and intensity non uniform artifact, due to the movement of the camera. This results in smooth and slowly varying change in image pixel values and lead to information loss, SNR gain and degradation of edge and finer details of image. Spatial filters are used for noise reduction. These filters may be linear or non-linear filters.

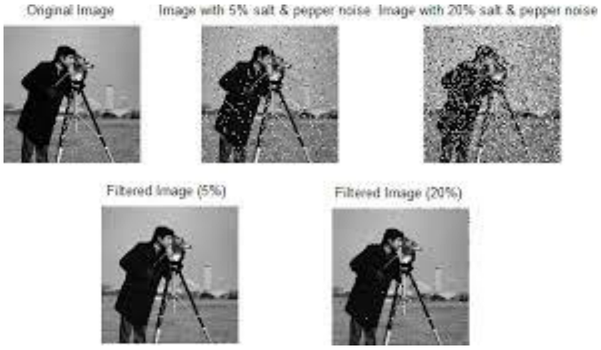

Original Image

Image with 5% salt & pepper noise

Image with 20% salt & pepper noise

Filtered Image (5%)

Filtered Image (20%)

Fig 3.2: Image Filtering

3.4.3 Image Segmentation

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as superpixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

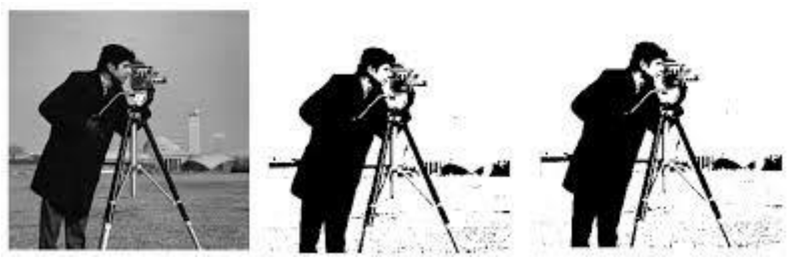


Fig 3.3 : image segmentation

3.4.4 Edge Detection

Edge detection is a basic tool used in image processing, basically for feature detection and extraction,which aim to identify points in a digital image where brightness Of image changes sharply and find discontinuities. The purpose of edge detection significantly reducing the data in an image and preserves the structural properties for further image processing.

Edge detection is basically image segmentation technique, divides spatial domain, on which the image is defined, into meaningful parts or regions. Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edges typically occur on the boundary between two different regions in an image.

Edge detection allows users to observe those features of an image where there is a more or less abrupt change in grey level or texture indicating the end of one region in the image and the beginning of another.

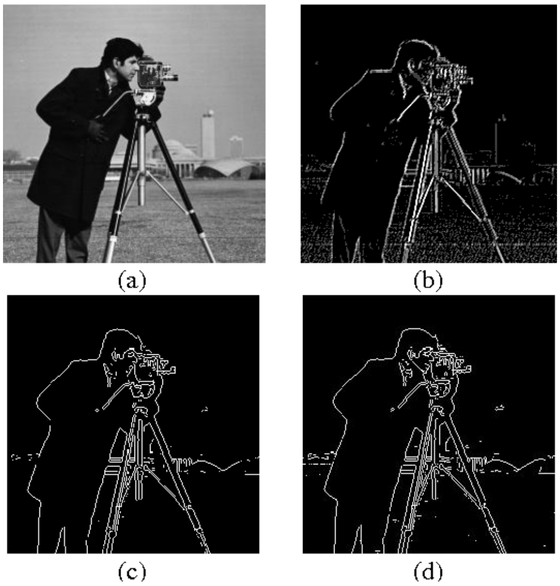

(a) (b) (c) (d)

Fig 3.4 : Edge Detection

3.4.5 Image Enhancement

The principal objective of image enhancement is to process a given image so that the result is more suitable than the original image for a specific application. It accentures or sharpens image features such as edges, boundaries, or contrast to make a graphic display e helpful for display and analysis. The enhancement doesn't increase the inherent information content of the data, but it increases the dynamic range of the chosen features so that they can be detected easily.

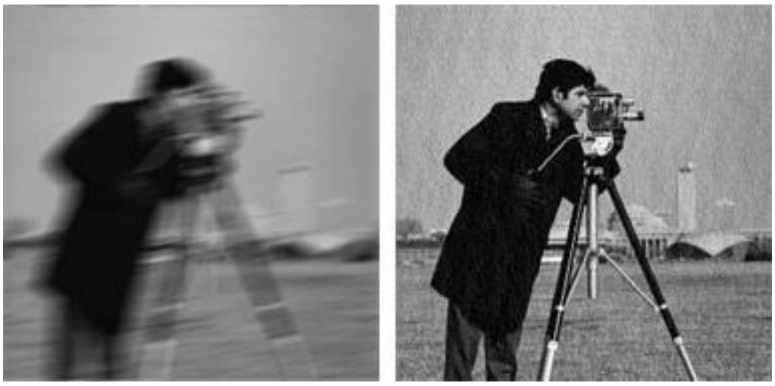


Fig 3.5 : Image Enhancement

3.5 APPLICATIONS OF IMAGE PROCESSING

1. Signature verification
2. Character recognition
3. Bio-metrics
4. Automatic target detection
5. Traffic monitoring

CHAPTER 4

FACE DETECTION USING HAAR CASCADE METHOD

4.1 INTRODUCTION

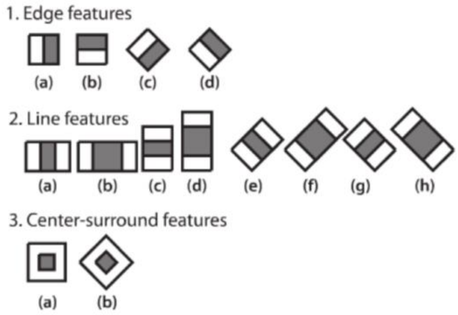
Face detection is a type of application classified under “computer vision” technology. It is the process in which algorithms are developed and trained to properly locate faces or objects (in object detection, a related system), in images. These can be in real time from a video camera or from photographs. An example where this technology is used are in airport security systems. In order to recognize a face, the camera software must first detect it and identify the features before making an identification. Likewise, when Facebook makes tagging suggestions to identify people in photos it must first locate the face. On social media apps like Snapchat, face detection is required to augment reality which allows users to virtually wear dog face masks using fancy filters. Another use of face detection is in smartphone face ID security.

In this project, I implemented a system for locating faces in digital images. These are in JPEG format only. Before we continue, we must differentiate between face recognition and face detection. They are not the same, but one depends on the other. In this case face recognition needs face detection for making an identification to “recognize” a face. I will only cover face detection.

Face detection uses classifiers , which are algorithms that detects what is either a face(1) or not a face(0) in an image. Classifiers have been trained to detect faces using thousands to millions of images in order to get more accuracy. OpenCV uses two types of classifiers, LBP (Local Binary Pattern) and Haar Cascades.

4.2 HAAR CASCADES

Haar Cascade classifier is based on the Haar Wavelet technique to analyse pixels in the image into squares by function. This uses “integral image” concepts to compute the “features” detected. Haar Cascades use the **Ada-boost** learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers then use cascading techniques to detect face in a image. Here are some Haar-Features



Face Detection determines the locations and sizes of human faces in arbitrary (digital) images.

In **Face Recognition**, the use of Face Detection comes first to determine and isolate a face before it can be recognized.

Fig 4.1: Haar feature extraction kerneals.

4.3 FEATURE EXTRACTION

As I mentioned earlier, Haar Cascades use machine learning techniques in which a function is trained from a lot of positive and negative images. This process in the algorithm is feature extraction.

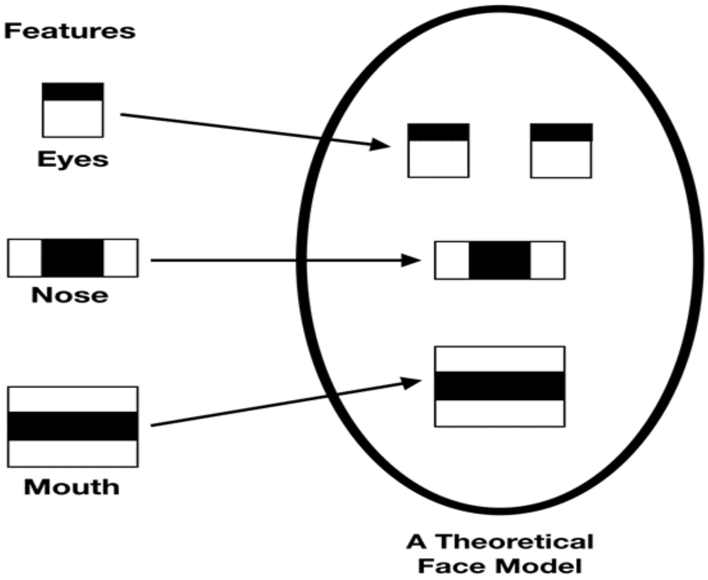
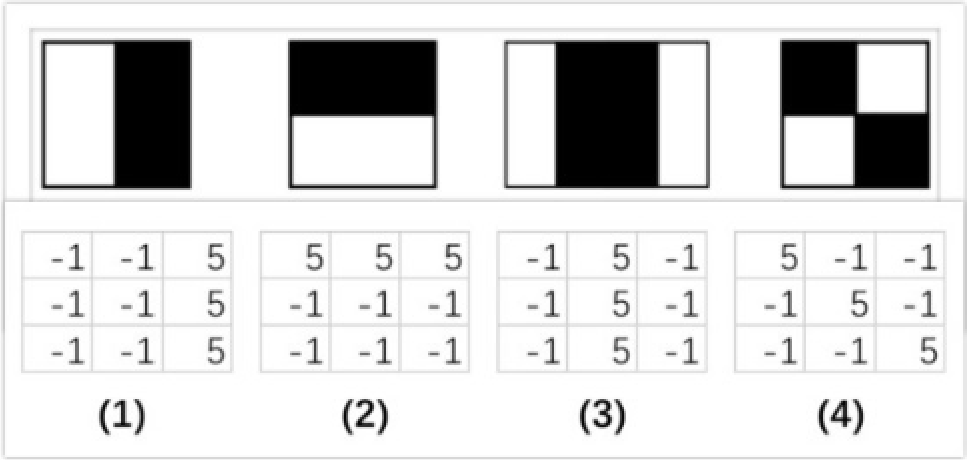


Fig 4.2: Face detection Kernels.

Haar features are similar to these convolution kernels which are used to detect the presence of that feature in the given image. The first two are “edge features”, used to detect edges. The third is a “line feature”, while the fourth is a “four rectangle feature”, most likely used to detected a slanted line.



-1	-1	5
-1	-1	5
-1	-1	5

(1)

5	5	5
-1	-1	-1
-1	-1	-1

(2)

-1	5	-1
-1	5	-1
-1	5	-1

(3)

5	-1	-1
-1	5	-1
-1	-1	5

(4)

Fig4.3: Kerneals

4.4 HAAR FEATURES APPLIED TO IMAGE

Each feature results in a single value which is calculated by subtracting the sum of pixels under white rectangle from the sum of pixels under black rectangle. Every haar feature have some sort of resemblance to identify a part of face.

Value = ∑ (pixels in white area) – ∑ (pixels in black area)

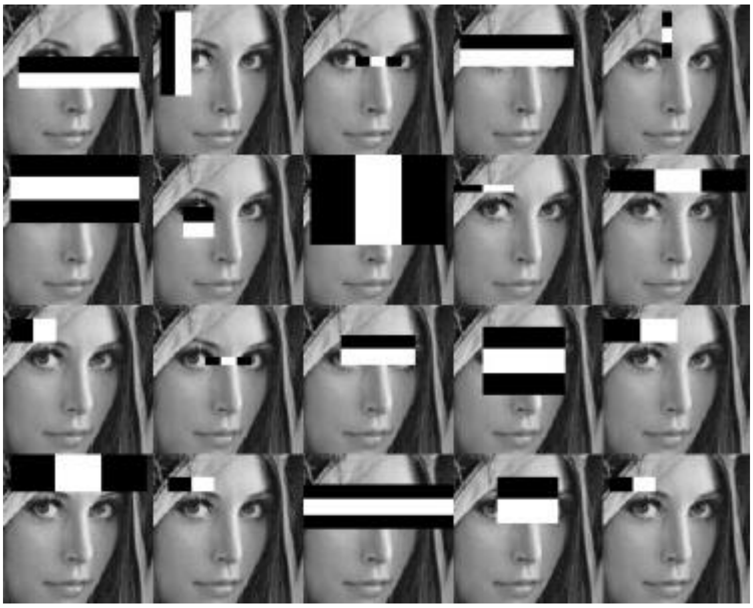


Fig 4.3: Detection of face in different parts

If we consider all possible parameters of the haar features like position, scale and type we end up calculating about 160,000+ features. So we need to evaluate huge set of features for every 24*24 PX.

So to avoid this we have a idea to avoid redundant features and pick only those features which are very useful for us. This can be done using AdaBoost.

4.5 ADABOOST

AdaBoost is a machine learning algorithm that able to find the best features among 160,000+ features. These features are also called as weak classifiers. After these features are found a weighted combination of all these features in used in evaluating and deciding any given window has a face or not. Each of the selected features (weak classifiers) are considered okay to be included if they can at least perform better than random guessing (detects more than half the cases). Each of the weak classifier is relevant detecting a part of face. Output of weak classifier is binary if it have identified a part of face or not.

Adaboost constructs a strong classifier as a linear combination of these weak classifiers.

Strong classifier = linear sum of weak classifiers

$F(x) = \sum (\alpha_i * f_i(x))$

here α_i are corresponding weights to each weak classifier $f_i(x)$

Even if there are more face's detected the excessive large amount of evaluated value remains negative. So the algorithm will more concentrate on discarding non faces and spending more time on probable face region. Hence a single strong classifier formed out of linear combination of all best features is not a good to evaluate on each window because of computation cost.

4.6 CASCADING

We start with simple classifiers which reject many of the negative sub-windows while detecting almost all positive sub-windows. Positive response from the first classifier triggers the evaluation of a second (more complex) classifier, and so on . A negative outcome at any point leads to the immediate rejection of the sub-window.

Given a input image sub-window we need to move our 24*24 window all over the image and compute 2,500 features for every window and take a linear combination of all outputs and see if it exceeds a certain threshold or not

- The basic principle of the Viola-Jones face detection algorithm is to scan the detector many times through the same image — each time with a new size.
- Even if an image should contain one or more faces it is obvious that an excessive large amount of the evaluated sub-windows will still be negatives (non-faces).
- So the algorithm should concentrate on discarding non-faces quickly and spend more on time on probable face regions.

- Hence a single strong classifier formed out of linear combination of all best features is not a good to evaluate on each window because of computation cost

Instead of calculating 2,500 features for every window we use idea of cascades. We do a sampling of 2,500 features into x different cascades. Now we can detect if there is face or not in different cascades linearly. If cascade$_i$ finds a face in a image then image is passed to next cascade. If no face is found is a cascade we can move to next window. This reduces the time complexity.

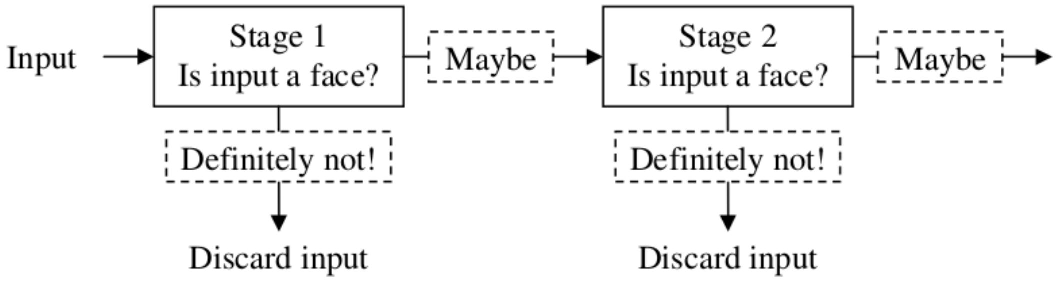
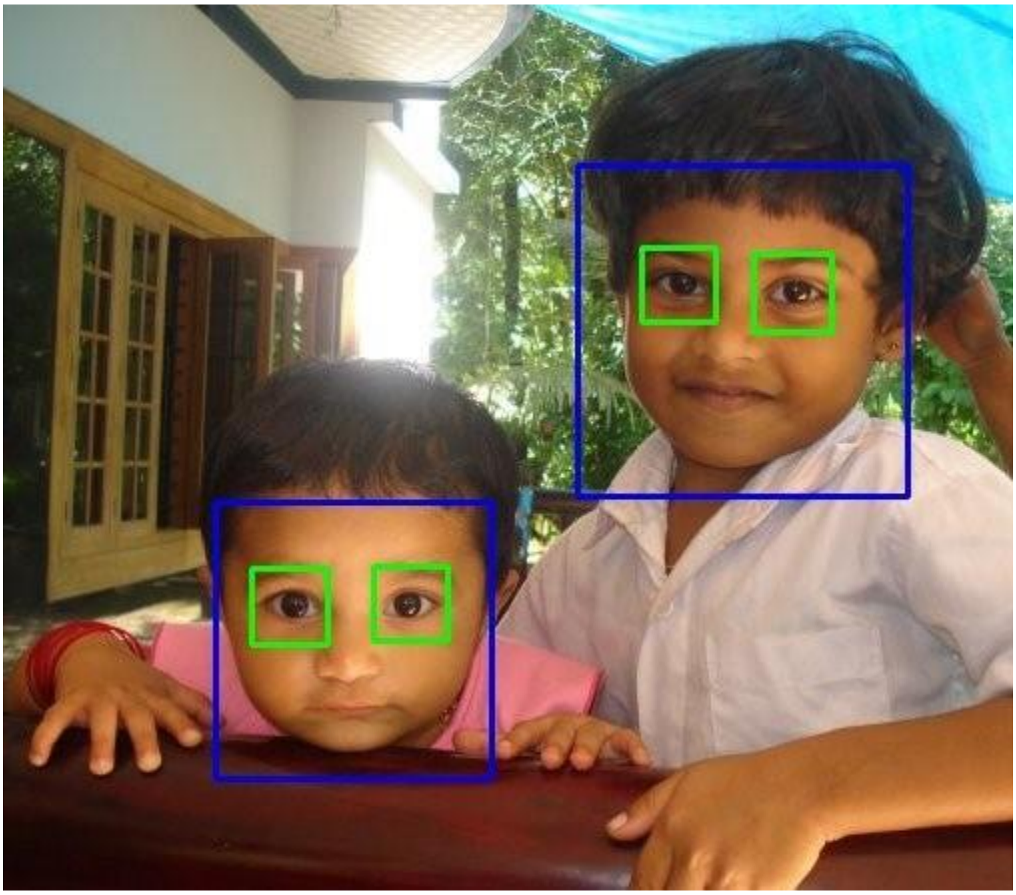


Fig 4.4: Cascading

The job of each stage is used to determine whether a given sub window is definitely not a face or may be a face. A given sub window is immediately discarded as not a face if it fails in any of the stage.

The output of face detection :



CHAPTER 5

HARDWARE COMPONENTS

5.1 RASPBERRY PI:

The **Raspberry Pi** is a series of small single-board computer developed in the United Kingdom by the Raspberry pi foundation to promote teaching of basic computer science in schools and in developing countries.The original model became far more popular than anticipated, selling outside its target market for uses such as robotics.

The Raspberry Pi 4 uses a Broadcom BCM2711 SoC with a 1.5 GHz 64-bit quad-core ARM Cortex A-72 processor, with 1MB shared L2 cache and 1 GB of RAM memory.

Raspberry pi, In terms of I/O, the Raspberry Pi 4 comes with a full-sized Ethernet port, two USB 3.0 ports, two USB 2.0 ports, two micro HDMI ports, USB type C port for power supply. The HDMI port can output a maximum resolution of 4K running at 30fps. With respect to wireless connectivity, the computer does feature 2.4 GHz and 5.0 GHz Wi-Fi network support with Bluetooth 5.0 to connected accessories.

The Raspberry Pi Foundation provides Raspbian, a Debian-based (32-bit) Linux distribution for download, as well as third-party Ubuntu, Windows 10 IoT core, RISC OS and other specialized distribution.

Raspberry pi is used in Home automation and educational purpose.Raspberry pi is affoardable and meets the requirements of our project.

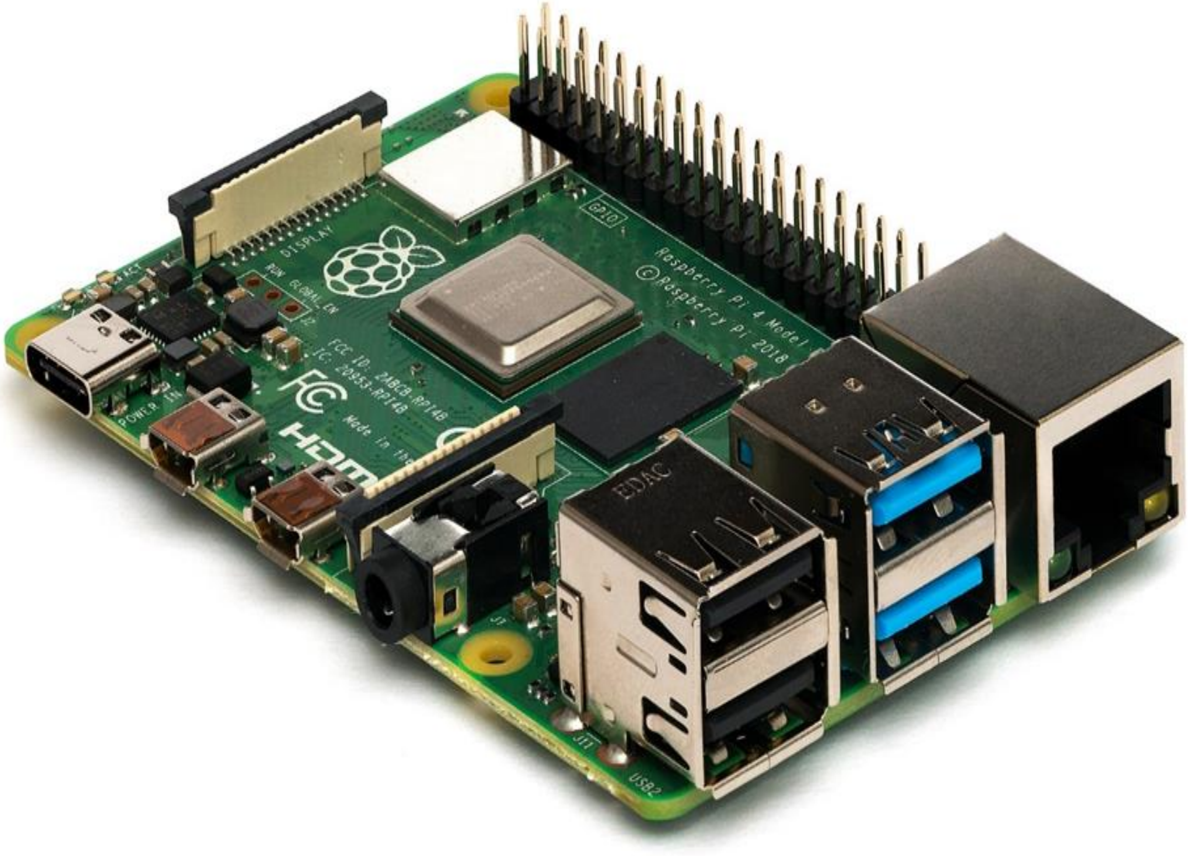


Fig:5.1: Raspberry Pi

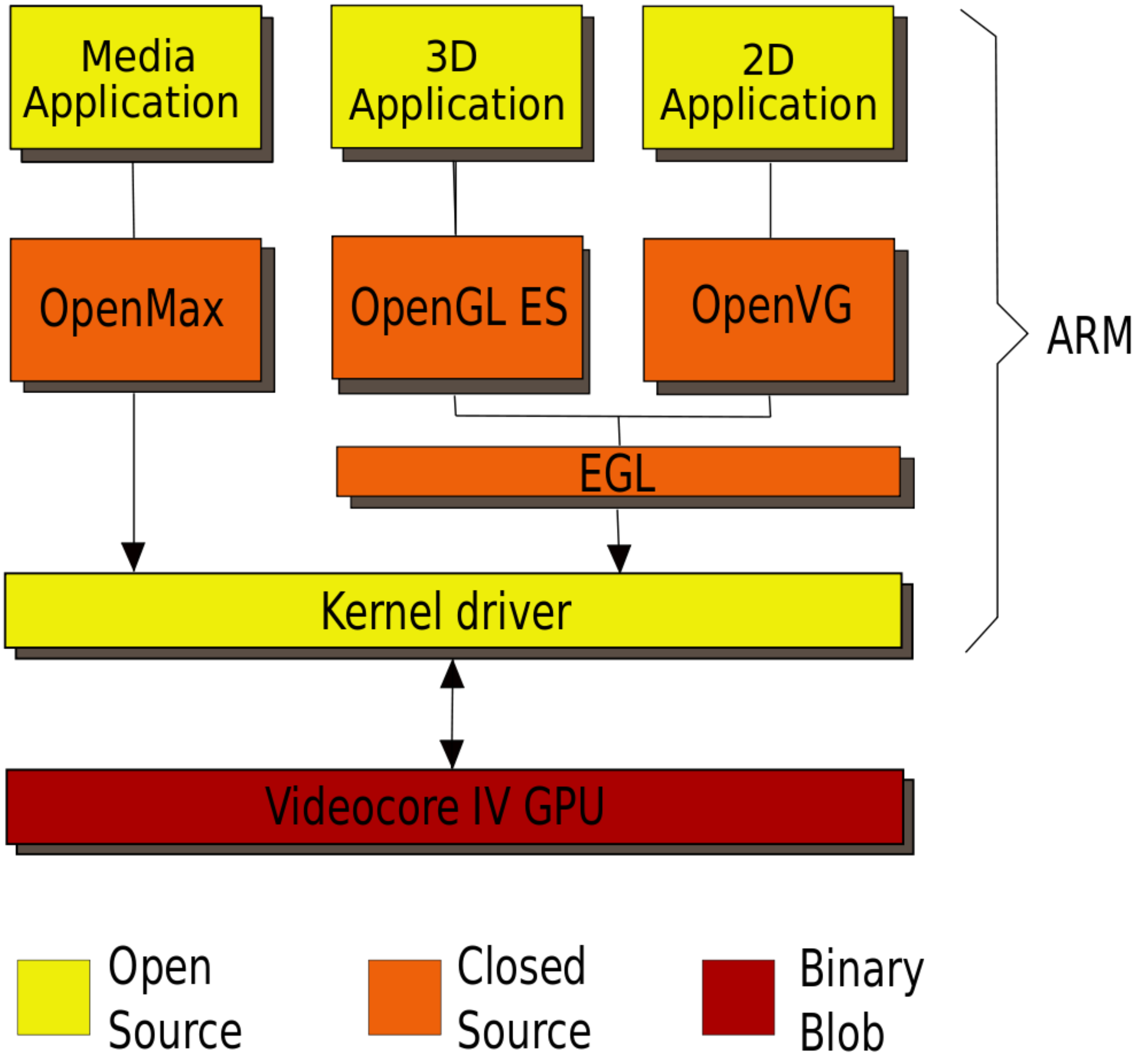
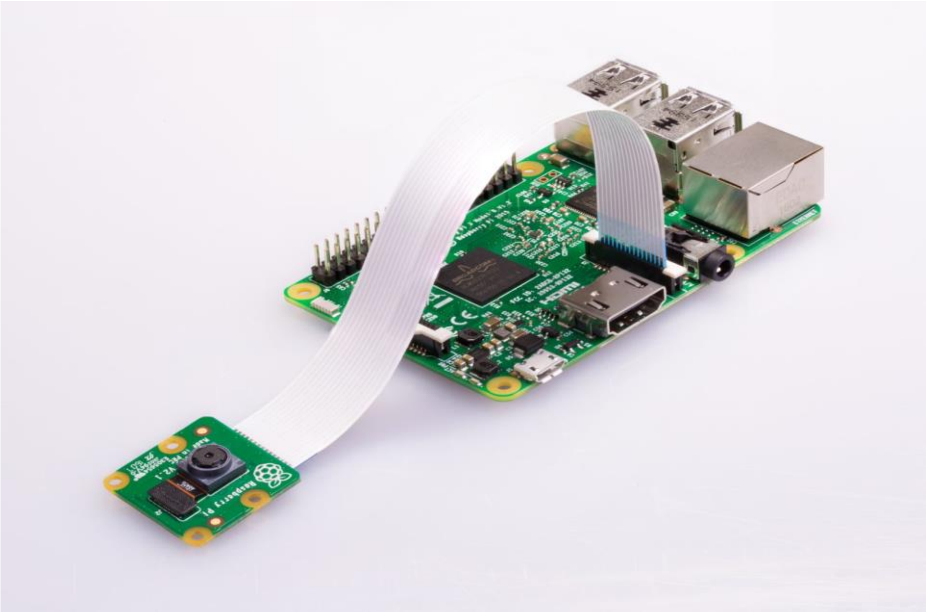


Fig:5.2: Kernel structure of Raspberry Pi

5.2 PI CAMERA:

The Raspberry Pi Camera Module is a 5 megapixel custom designed add-on for Raspberry Pi, featuring a fixed focus lens. It's capable of 2592 x 1944 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras. 5 megapixel native resolution sensor-capable of 2592 x 1944 pixel static images Supports 1080p30, 720p60 and 640x480p60/90 video Camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system The board itself is tiny, at around 25mm x 20mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable. The sensor itself has a native resolution of 5 megapixel, and has a fixed focus lens on-board. In terms of still images, the camera is capable of 2592 x 1944 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. The camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system. 1.4 m X 1.4 m pixel with Omni BSI technology for high performance (high sensitivity, low crosstalk, low noise) optical size of 1/4" automatic image control functions: - automatic exposure control (AEC) - automatic white balance (AWB) - automatic band filter (ABF) - automatic 50/60 Hz luminance detection - automatic black level calibration (ABLC) programmable controls for frame rate , AEC/AGC 16-zone size/position/weight control, mirror and flip, cropping, windowing, and panning digital video port (DVP) parallel output interface 32 bytes of embedded one-time programmable (OTP) memory.

The Raspberry pi camera module is small yet has a good capture resolution and affordable which makes it very suitable for its implementation in the home automation technologies and is being utilized in the current project as the primary image sensor which captures the face of the user



.

Fig5.3: Raspberry pi with Pi Camera

5.3 Raspberry PI OLED display:

An **organic light-emitting diode** (**OLED** or **Organic LED**), also known as an **organic EL** (organic electroluminescent) diode,is a light emitting diode(LED) in which the emissive electroluminescent layer is a film of organic compound that emits light in response to an electric current. This organic layer is situated between two electrodes; typically, at least one of these electrodes is transparent. OLEDs are used to create digital displays in devices such as television screens, computer moniters, portable systems such as smart phones, hand held game consoles etc.

In our project we utilize 12C OLED display for the display of instructions and to communicate with the user.According to the application and budget an LCD or LED panel can also be utilized to meet the requirements of the user.

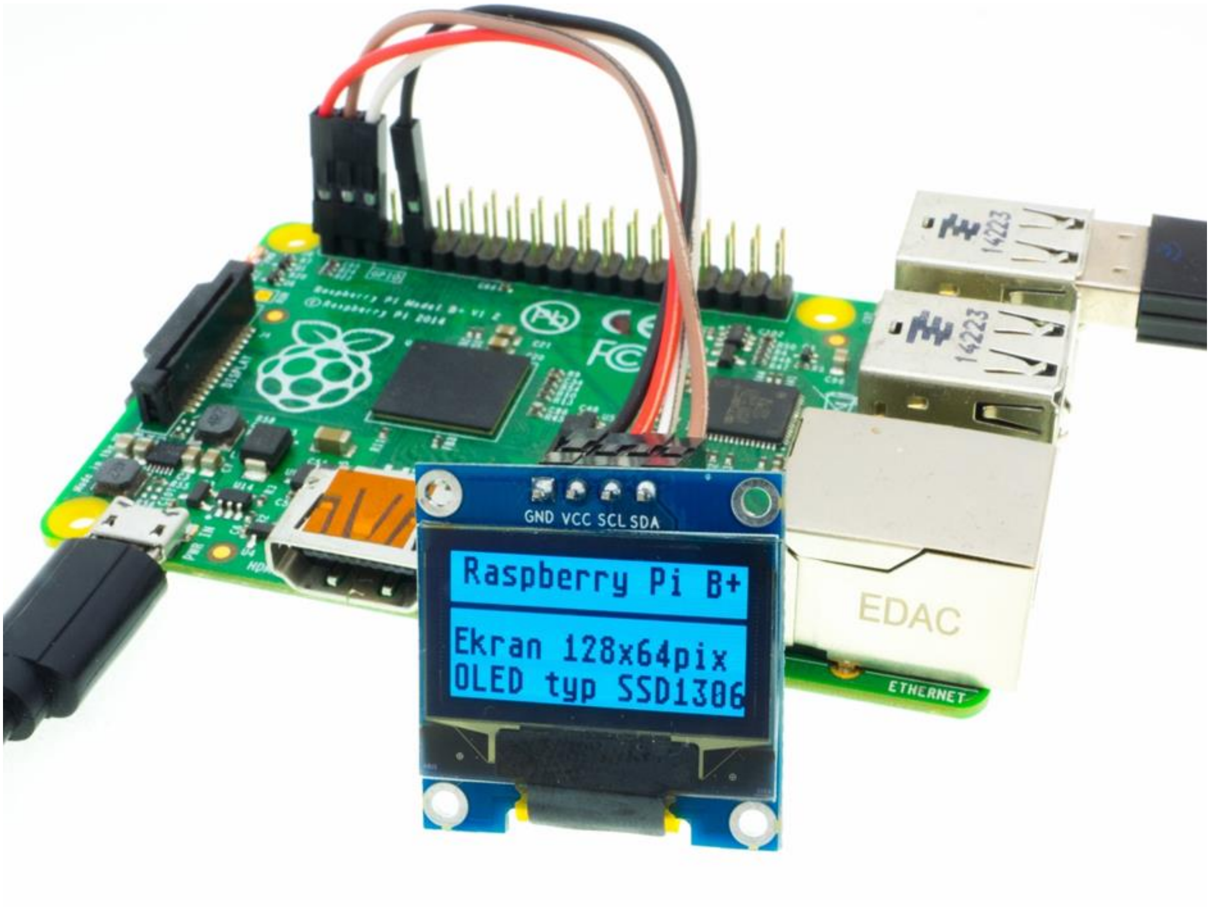


Fig 5.4: Raspberry pi with OLED display

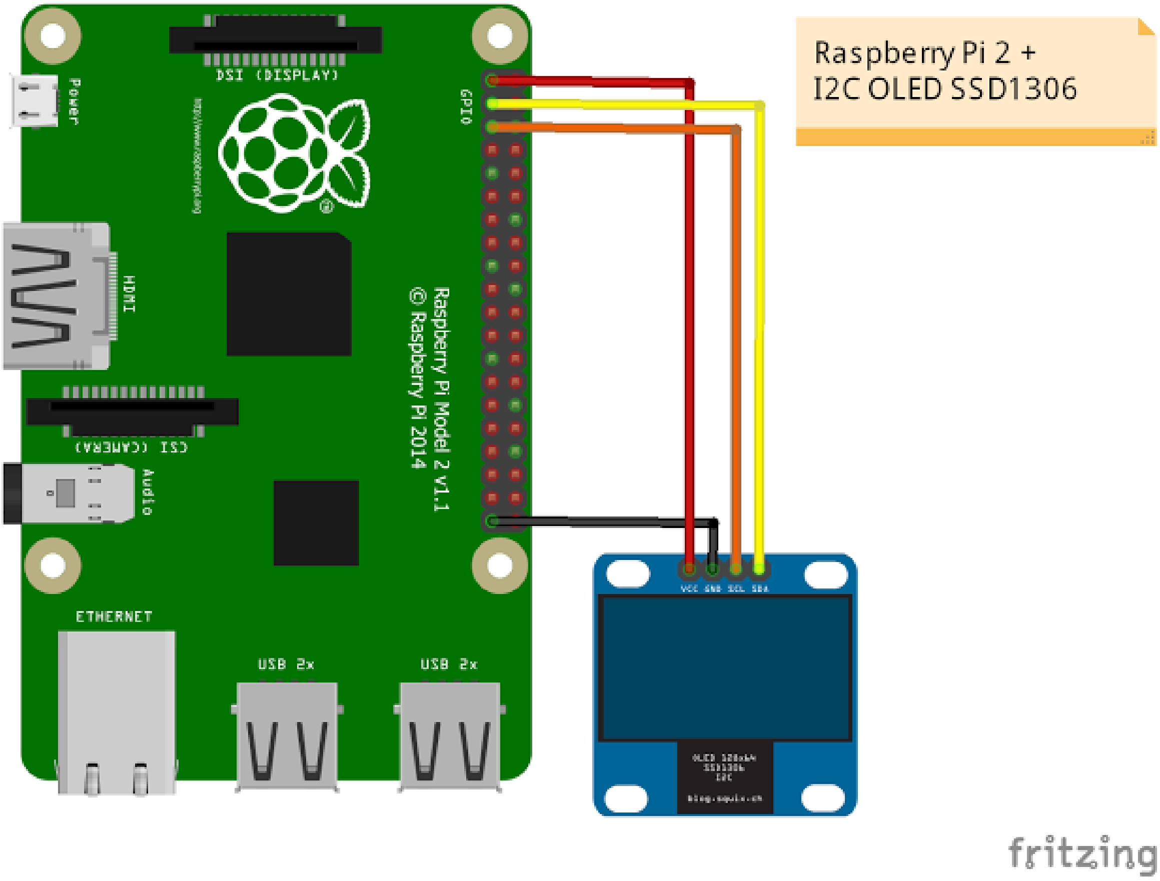


Fig 5.5: Raspberry pi with OLED circuit

CHAPTER 6

OPEN CV

6.1 INTRODUCTION

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for

computer vision applications and to accelerate the use of machine perception in the commercial products. OpenCV was originally developed by Intel back in 1999 and is now maintained by Willow Garage and It See. OpenCV (Open Source Computer Vision) is a library of programming functions for realtime computer vision, itt uses a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java (Android) interfaces and supports Windows, Linux, Android, iOS and Mac OS. It has more than 2500 optimized algorithms. Adopted all around the world, OpenCV has more than 9 million downloads growing by nearly 200K/month. Usage ranges from interactive art, to mines inspection, stitching maps on the web on through advanced robotics.

6.2 HISTORY

Officially launched in 1999 the OpenCV project was initially an Intel Research initiative to advance CPU -intensive applications, part of a series of projects including real-time ray tracing and 3D display walls. The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel's Performance Library Team. In the early days of OpenCV, the goals of the project were described as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.

- Advance vision-based commercial applications by making portable performance-optimized code available for free – with a license that did not require code to be open or free itself.

The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008.

The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the C++ interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six months and development is now done by an independent Russian team supported by commercial corporations.

In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer and user site.

On May 2016, Intel signed an agreement to acquire Itseez,[1] a leading developer of OpenCVOpenCV is a cross-platform library using which we can develop real-time **computer vision applications**. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

Let’s start the chapter by defining the term "Computer Vision".

6.2.1 Computer Vision

Computer Vision can be defined as a discipline that explains how to reconstruct, interrupt, and understand a 3D scene from its 2D images, in terms of the properties of the structure present in the scene. It deals with modeling and replicating human vision using computer software and hardware.

Computer Vision overlaps significantly with the following fields −

- **Image Processing** − It focuses on image manipulation.
- **Pattern Recognition** − It explains various techniques to classify patterns.
- **Photogrammetry** − It is concerned with obtaining accurate measurements from images.

Computer Vision Vs Image Processing

Image processing deals with image-to-image transformation. The input and output of image processing are both images.

Computer vision is the construction of explicit, meaningful descriptions of physical objects from their image. The output of computer vision is a description or an interpretation of structures in 3D scene.

Features of OpenCV Library

Using OpenCV library, you can −

- Read and write images
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or images.
- Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

OpenCV was originally developed in C++. In addition to it, Python and Java bindings were provided. OpenCV runs on various Operating Systems such as windows, Linux, OSx, FreeBSD, Net BSD, Open BSD, etc.

This tutorial explains the concepts of OpenCV with examples using Java bindings.

OpenCV Library Modules

Following are the main library modules of the OpenCV library.

Core Functionality

This module covers the basic data structures such as Scalar, Point, Range, etc., that are used to build OpenCV applications. In addition to these, it also includes the multidimensional array **Mat**, which is used to store the images. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.core**.

Image Processing

This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.imgproc**.

Video

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.video**

Installing OpenCV on a Raspberry Pi

To install OpenCV for Python, we need to have Python installed. Since Raspberry Pi's are preloaded with Python, we can install OpenCV directly.

First we need to make sure that Raspberry pi is up to date. Type the following command to update your Raspberry Pi.

```
    sudo apt-get update
```

Then type the following command in the terminal to get the dependencies required for installing OpenCV on your Raspberry Pi.

```
    sudo apt-get install libhdf5-dev -y && sudo apt-get install libhdf5-serial-dev -y &&
sudo apt-get install libatlas-base-dev -y && sudo apt-get install libjasper-dev -y && sudo
apt-get install libqtgui4 -y && sudo apt-get install libqt4-test -y
```

Now we can install OpenCV on Raspberry Pi. Type the following command to install OpenCV 4 for Python 3 on your Raspberry Pi, **pip3** tells us that OpenCV will get installed for Python 3.

```
pip3 install opencv-contrib-python==4.1.0.25
```

After those steps, OpenCV should be installed. Let's test our work!

6.3 TESTING OPEN CV

To check whether OpenCV is correctly installed or not, try importing OpenCV by typing:

```
python3
```

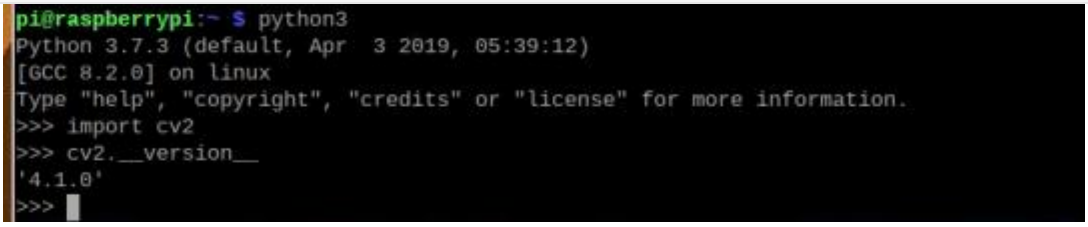
then:

```
import cv2
```

If no errors are shown, your installation was successful!

To know which version of OpenCV you have, type the following command:

```
cv2.__version__
```



```
pi@raspberrypi:~ $ python3
Python 3.7.3 (default, Apr  3 2019, 05:39:12)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.1.0'
>>> 
```

This message tells you which version on OpenCV you installed on your Raspberry Pi.

Recommended Optional Libraries

There are also other libraries to install that will come in handy when you do object detection and other projects, so I highly recommend you to also install these.

NumPy

The first library is NumPy — a library that makes array operations in Python easy to perform. Install NumPy by typing the following command:

```
pip3 install python-numpy
```

Matplotlib

The second library is Matplotlib. Matplotlib is a Python plotting library that produces publication quality figures in a variety of hard copy formats and interactive environments across platforms. Install Matplotlib by typing the following command:

```
pip3 install python-matplotlib
```

Now we are done installing OpenCV and helpful accompanying libraries on Raspberry Pi. Let's move forward to face detection using OpenCV on Raspberry Pi.

Computer Vision Advantages

Now, in the OpenCV tutorial, let’s learn about the benefits of computer vision:

- **Open source:** The source code of OpenCV is free to modify and customize according to individual requirements.
- **Fast:** Originally OpenCV is written in C++. Its performance is as fast as C++, the Python wrappers use C++ code in the background.
- **Easier to integrate:** OpenCV makes use of numpy arrays, which are efficient for performing operations and data can be used with other libraries like matplotlib and scikit-learn.
- **Ease of coding:** Python is easier to code as compared to other languages. We don’t need to take care of pointers, memory management, etc.
- **Fast prototyping:** With Python OpenCV library, we can quickly build prototypes. Python is concise, clean and shorter code is needed to perform some tasks with other languages.

6.4 APPLICATIONS OF COMPUTER VISION

Here we have listed down some of major domains where Computer Vision is heavily used.

Robotics Application

- Localization − Determine robot location automatically
- Navigation
- Obstacles avoidance
- Assembly (peg-in-hole, welding, painting)
- Manipulation (e.g. PUMA robot manipulator)
- Human Robot Interaction (HRI) − Intelligent robotics to interact with and serve people

Medicine Application

- Classification and detection (e.g. lesion or cells classification and tumor detection)
- 2D/3D segmentation
- 3D human organ reconstruction (MRI or ultrasound)
- Vision-guided robotics surgery

Industrial Automation Application

- Industrial inspection (defect detection)
- Assembly
- Barcode and package label reading
- Object sorting
- Document understanding (e.g. OCR)

Security Application

- Biometrics (iris, finger print, face recognition)
- Surveillance − Detecting certain suspicious activities or behaviors

Transportation Application

- Autonomous vehicle

Safety, e.g., driver vigilance monitorin

6.4.1 Event Detection

CCTV cameras are everywhere around us in offices, roads, hospitals, banks, train stations, parking lots, etc and 24/7 surveillance is difficult. The computer vision techniques allow us to monitor the events in real-time and detect anomalies or any specific action detection. Automatic Number Plate Recognition (ANPR) system can be used to control automatic gates, vehicle tracking, analyzing crowd and counting number of people.

Industrial Automation

Automatic inspection of objects and classifying them into different categories play a major role in manufacturing industries. Industrial robots use computer vision algorithms to perform various tasks like separating two or more different objects into their respective categories, detecting whether the product is labeled or not. If not, then we can reject the product on the conveyor belt.

6.4.2 Medical Image Processing

The progress in computer vision has led to extensive use of the medical imaging data to provide us with better prediction, diagnosis, and treatment of diseases. Some examples of this are the detection of tumors, arteriosclerosis or other malign changes, measurement of organ dimensions, blood flow, enhancement of ultrasonic or X-ray images that are interpreted by humans.

6.4.3 Self-driving Vehicles

Artificial intelligence is on the boom right now and companies are investing in self-driving technology. Computer vision is used to detect lanes and find a path for autonomous vehicles. Information from various sensors is analyzed and used to detect objects on the path like traffic lights, traffic signs and according to obstacles, we decide the appropriate action that needs to be taken.

6.4.4 Military Applications

The military is probably one of the largest areas for computer vision. More advanced systems for missile guidance are developed to make dynamic decisions based on the information provided by various sensors including image sensors. Modern military concepts are emerging like “Battlefield awareness” in which strategic decisions are made after analyzing the information provided by sensors, detecting enemy soldiers or vehicles.

CHAPTER 7

METHOLOGY

7.1 INTRODUCTION

The most important of feature of any home security system is to detect the people who enter or leave the house. Instead of monitoring that through passwords or pins unique faces can be made use of as they are one’s biometric trait. These are innate and cannot be modified or stolen easily. The level of security can be raised by using face detection.

The proposed face recognition door lock security system has been developed to prevent robbery in highly secure areas like home environment with lesser power consumption and more reliable standalone security device for both Intruder detection and for door security.

This system is powered by raspberry pi circuit. Raspberry Pi electronic board is operated on Battery power supply, wireless internet connectivity by using USB modem, it includes camera, PIR motion sensor and a door. Whenever the person comes in front of the door, it recognizes the face and if it is registered then it unlocks the door, if the face is not registered it will raise an alarm and clicks a picture and send it on the registered number. This is how the system works.

The overall view of how the components work:

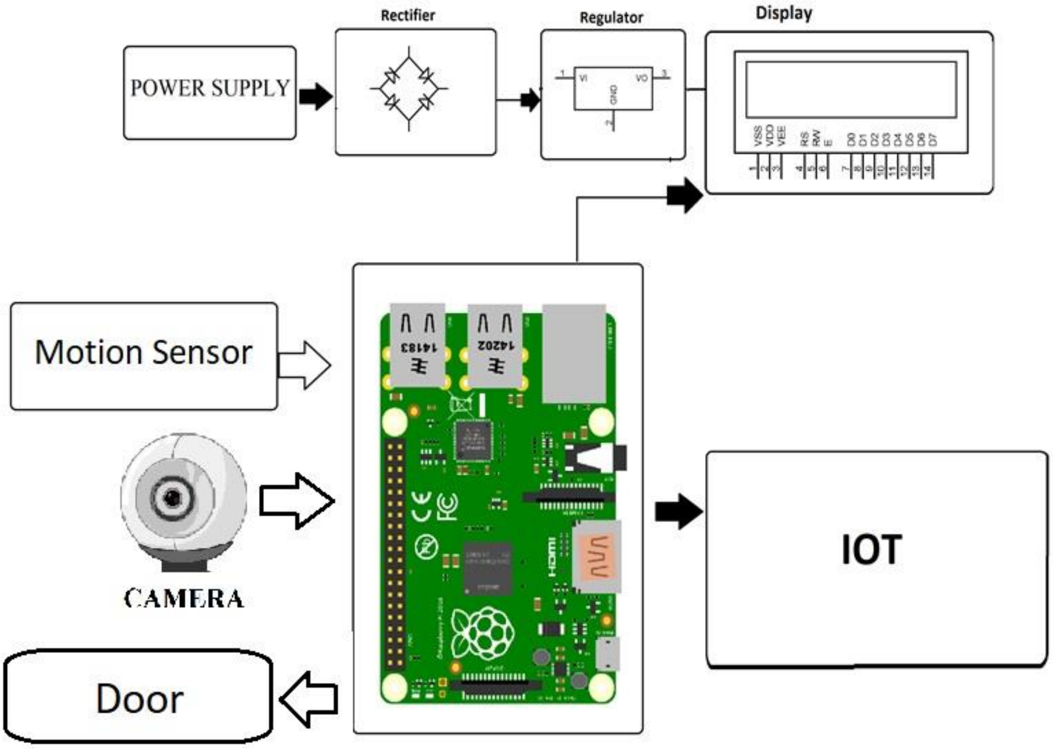


Fig 7.1: Rough Circuit Diagram

Methodology mainly involves three basic steps :

1) Data gathering

2) Training the recognizer

3) Recognition

Data gathering :

Facial Recognition Data Collection is helpful in collecting information like bridge of the nose, contour of the lips, ears, chin as well as spacing of the eyes. With the help of this technology these automated systems can be used to check or identify the identity of personalities in just a few seconds based on their facial features. This can even be done in the middle of a crowd and within unstable and dynamic environments.

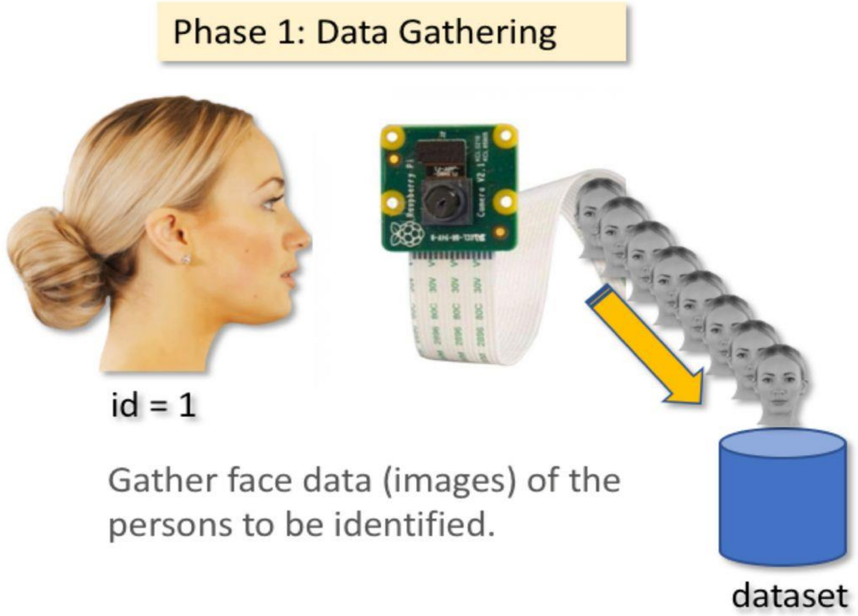


Fig7.2: Data Aquisition

Training the recognizer :

To perform face recognition we need to train a face recognizer, using a pre labeled dataset, In my previous post we created a labeled dataset for our face recognition system, now its time to use that dataset to train a face recognizer using opencv python, First create a python “trainner.py” file in the same folder where we saved out dataset generator script in the previous post, and then create a folder in the same directory name it “trainner”, this is the folder where we are going to save our recognizer after training.

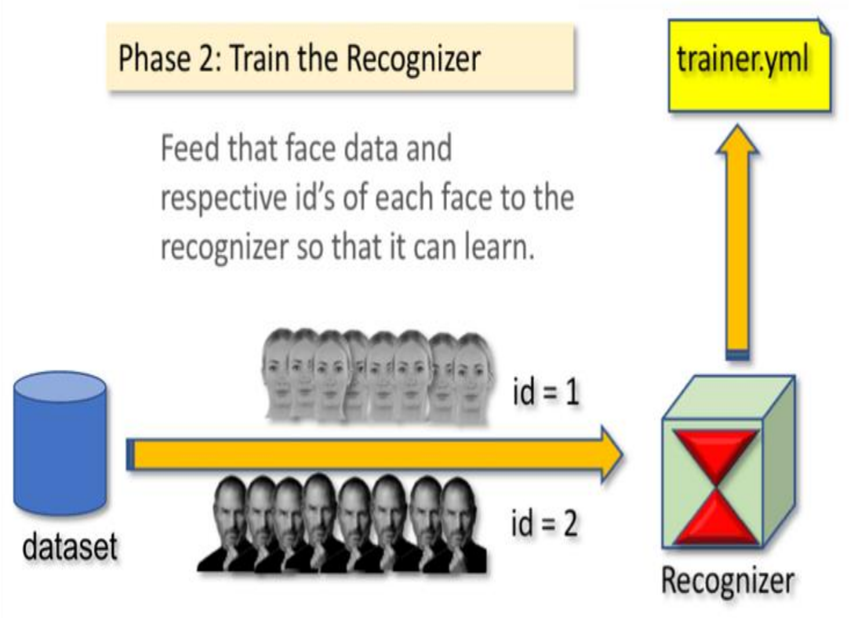

Phase 2: Train the Recognizer
trainer.yml
Feed that face data and respective id's of each face to the recognizer so that it can learn.
id = 1
id = 2
dataset
Recognizer

Fig7.3: Training

Recognition:

In recognition process the data of the user which is present in the data base is compared with the person data who is in front of the camera both data is compared by local binary patter histograms of two data if it matches then the person is recognised and go to further proces. Else it displays unknow I'd

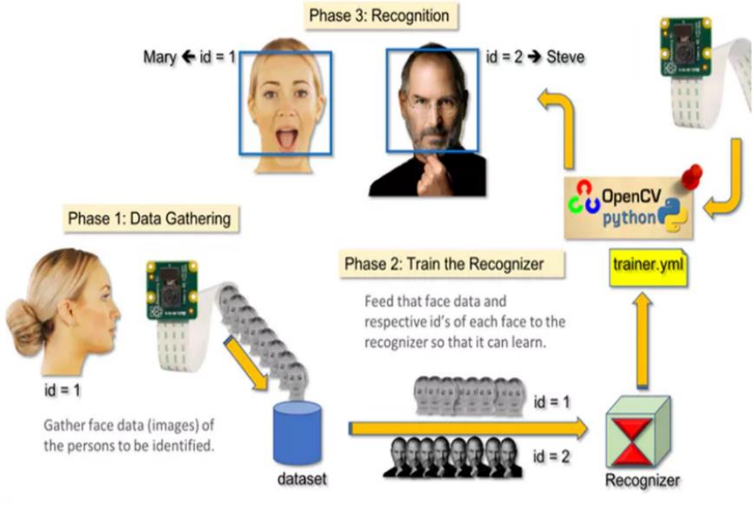
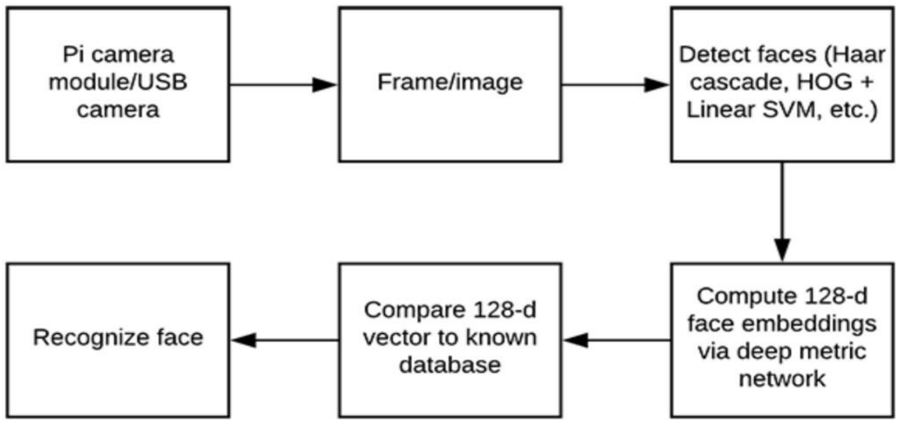

Phase 3: Recognition
Mary ← id = 1
id = 2 → Steve
OpenCV
python
Phase 1: Data Gathering
id = 1
Gather face data (images) of the persons to be identified.
dataset
Phase 2: Train the Recognizer
Feed that face data and respective id's of each face to the recognizer so that it can learn.
trainer.yml
id = 1
id = 2
Recognizer

Fig7.4: Recognition

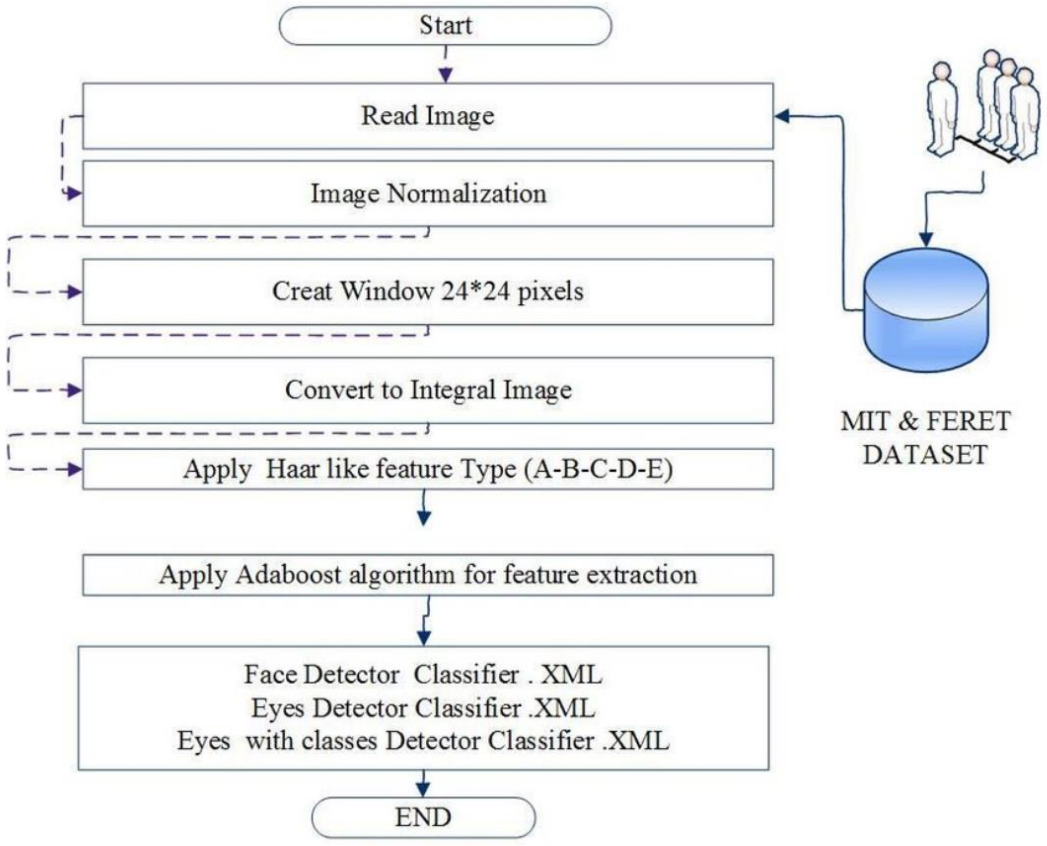
7.2 BLOCK DIAGRAM



Pi camera module/USB camera
Frame/image
Detect faces (Haar cascade, HOG + Linear SVM, etc.)
Compute 128-d face embeddings via deep metric network
Compare 128-d vector to known database
Recognize face

Fig7.5: Block diagram

7.3 FLOW CHART



Start
Read Image
Image Normalization
Creat Window 24*24 pixels
Convert to Integral Image
Apply Haar like feature Type (A-B-C-D-E)
Apply Adaboost algorithm for feature extraction
Face Detector Classifier . XML
Eyes Detector Classifier .XML
Eyes with classes Detector Classifier .XML
END
MIT & FERET DATASET

Fig7.6: Flow chart of methodology

In this we will learn how we can build our own Face Recognition system using the OpenCV Library on Raspberry Pi. The advantage of installing this system on portable Raspberry Pi is that you can install it anywhere to work it as surveillance system. Like all Face Recognition systems, the tutorial will involve two python scripts, one is a Trainer program which will analyze a set of photos of a particular person and create a dataset (YML File). The second program is the Recognizer program which detects a face and then uses this YML file to recognize the face and mention the person name. Both the programs that we will discuss here are for Raspberry Pi (Linux), but will also work on Windows Computers with very slight changes. We already have series of Tutorials for beginners for getting started with OpenCV, you can check all the OpenCV .

As we know using the OpenCV Library to detect and recognize faces. So make sure you to install OpenCV Library on Pi before proceeding with this tutorial. Also Power your Pi with a 2A adapter and connect it to a display monitor via HDMI cable since we will not be able to get the video output through SSH

Before we start, it is important to understand that Face Detection and Face Recognition are two different things. In Face Detection only the Face of a person is detected the software will have no Idea who that Person is. In Face Recognition the software will not only detect the face but will also recognize the person. Now, it should be clear that we need to perform Face Detection before performing Face Recognition. It would not be possible for me to explain how exactly OpenCV detects a face or any other object for that matter.

A video feed from a webcam is nothing more than a long sequence of still images being updated one after the other. And each of these images is just a collection of pixels of different values put together in its respective position. So how can a program detect a face from these pixels and further recognize the person in it? There are a lot of algorithms behind it and trying to explain them is beyond the scope of this article, but since we are using the OpenCV library it very simple to perform face Recognition without getting deeper into the concepts.

Face Detection using Cascade Classifiers in OpenCV

Only if we are able to detect a face we will able to recognize it or remember it. To detect an object such as face OpenCV uses something called Classifiers. These Classifiers are pre-trained set of data (XML File) which can be used to detect a particular object in our case a face. You can learn more about Face Detection Classifiers here. Apart from detecting Face, Classifiers can detect other objects like nose, eyes, Vehicle License Plate, Smile etc. The list of Case Classifiers can be downloaded from the ZIP file below

Classifiers for Object detection in Python

Alternatively OpenCV also allows you to create your own Classifier which can be used to detect any other object in an Image by Training your Cascade Classifier. In this tutorial we will use a classifier called “haarcascade_frontalface_default.xml” which will detect the face from front position. We will see more on how to use Classifiers in the Programming section.

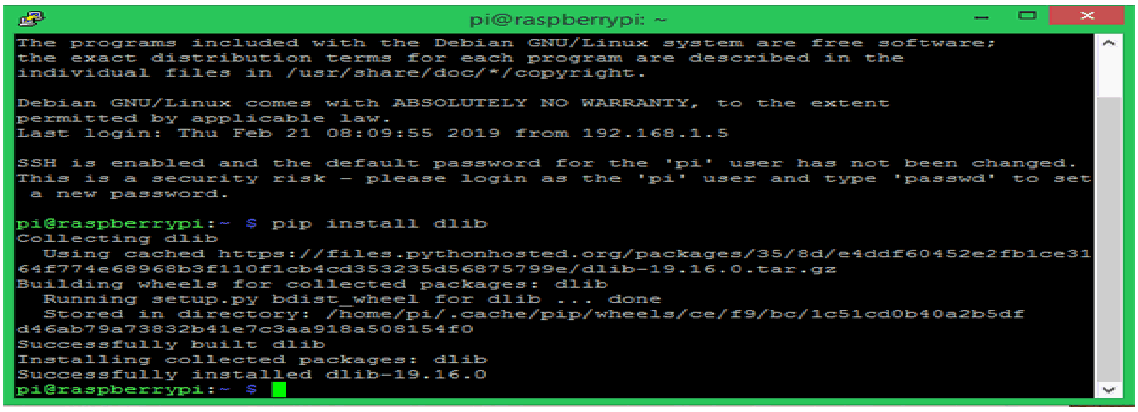
Installing the Required Packages

Make sure pip is installed and then proceed with installing the following packages.

Install dlib: Dlib is a toolkit for real world Machine Learning and data analysis applications. To install dlib, just enter the following command in the terminal

Pip install dlib

This should install dlib and when successful you will get a screen like this.



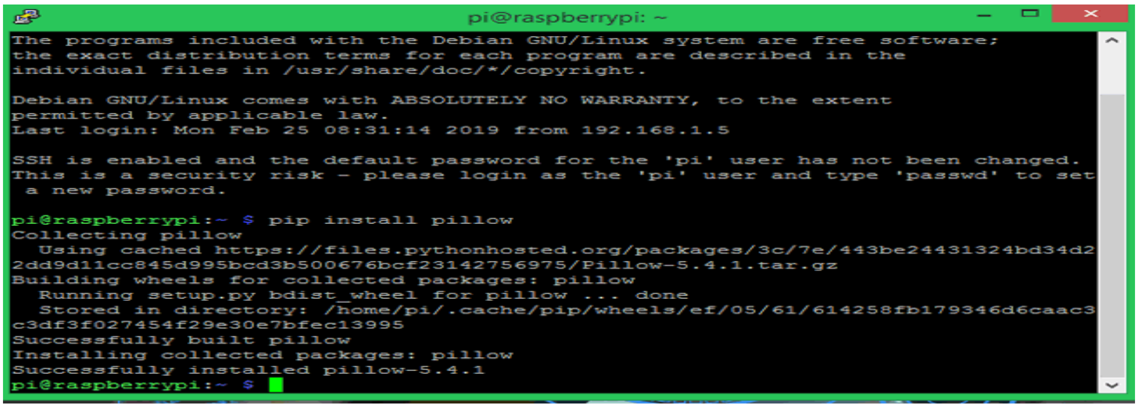
```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb 21 08:09:55 2019 from 192.168.1.5

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi:~ $ pip install dlib
Collecting dlib
  Using cached https://files.pythonhosted.org/packages/35/8d/e4ddf60452e2fb1ce31
64f774e68968b3f110f1cb4cd353235d56875799e/dlib-19.16.0.tar.gz
Building wheels for collected packages: dlib
  Running setup.py bdist_wheel for dlib ... done
  Stored in directory: /home/pi/.cache/pip/wheels/ce/f9/bc/1c51cd0b40a2b5df
d46ab79a73832b41e7c3aa918a508154f0
Successfully built dlib
Installing collected packages: dlib
Successfully installed dlib-19.16.0
pi@raspberrypi:~ $
```

Fig7. 7: Installing dlib

Install pillow: Pillow also known as PIL stands for Python Imaging Library which is used to open, manipulate and save images in different format. To install PIL use the following command

Pip install pillow

Once installed you will get a success message as shown below



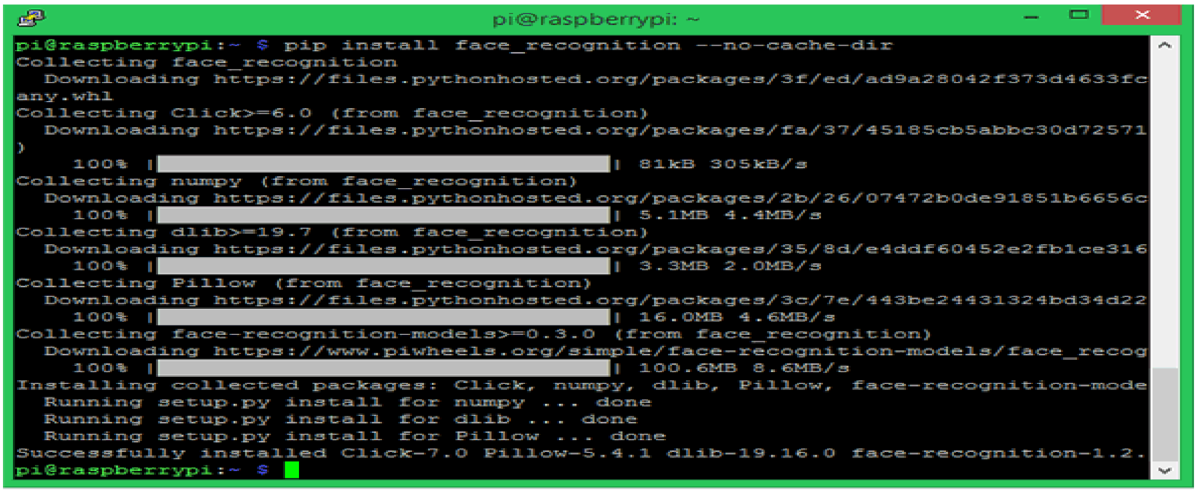
```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 25 08:31:14 2019 from 192.168.1.5

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi:~ $ pip install pillow
Collecting pillow
  Using cached https://files.pythonhosted.org/packages/3c/7e/443be24431324bd34d2
2dd9d11cc845d995bcd3b500676bcf23142756975/Pillow-5.4.1.tar.gz
Building wheels for collected packages: pillow
  Running setup.py bdist_wheel for pillow ... done
  Stored in directory: /home/pi/.cache/pip/wheels/ef/05/61/614258fb179346d6caac3
c3df3f027454f29e30e7bfec13995
Successfully built pillow
Installing collected packages: pillow
Successfully installed pillow-5.4.1
pi@raspberrypi:~ $
```

Fig7. 8: Install Pillow

Install face_recognition: The face_recognition library for python is considered to be simplest library to recognize and manipulate faces. We will be using this library to train and recognize faces. To install this library follow the command

Pip install face_recognition –no –cache-dir

When installed successfully you should see a screen like shown below. The library is heavy and most people will face memory exceeding problems hence I have used the "—no –cache-dir" code to install the library without saving the cache files.

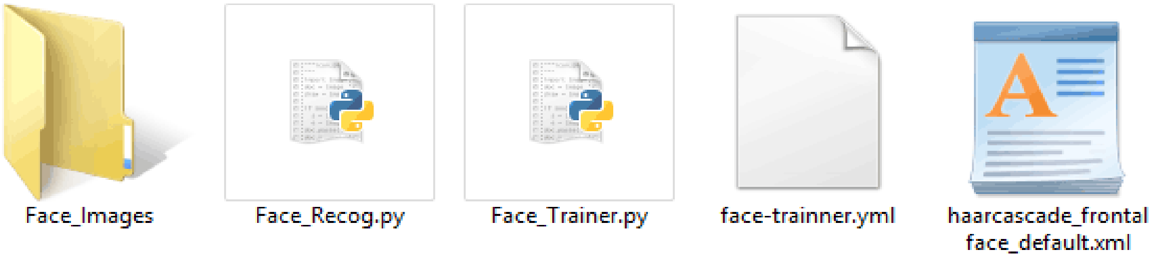


```
pi@raspberrypi:~ $ pip install face_recognition --no-cache-dir
Collecting face_recognition
  Downloading https://files.pythonhosted.org/packages/3f/ed/ad9a28042f373d4633fc
any.whl
Collecting Click>=6.0 (from face_recognition)
  Downloading https://files.pythonhosted.org/packages/fa/37/45185cb5abbc30d72571
)
    100% |████████████████████████████████| 81kB 305kB/s
Collecting numpy (from face_recognition)
  Downloading https://files.pythonhosted.org/packages/2b/26/07472b0de91851b6656c
    100% |████████████████████████████████| 5.1MB 4.4MB/s
Collecting dlib>=19.7 (from face_recognition)
  Downloading https://files.pythonhosted.org/packages/35/8d/e4ddf60452e2fb1ce316
    100% |████████████████████████████████| 3.3MB 2.0MB/s
Collecting Pillow (from face_recognition)
  Downloading https://files.pythonhosted.org/packages/3c/7e/443be24431324bd34d22
    100% |████████████████████████████████| 16.0MB 4.6MB/s
Collecting face-recognition-models>=0.3.0 (from face_recognition)
  Downloading https://www.piwheels.org/simple/face-recognition-models/face_recog
    100% |████████████████████████████████| 100.6MB 8.6MB/s
Installing collected packages: Click, numpy, dlib, Pillow, face-recognition-mode
  Running setup.py install for numpy ... done
  Running setup.py install for dlib ... done
  Running setup.py install for Pillow ... done
Successfully installed Click-7.0 Pillow-5.4.1 dlib-19.16.0 face-recognition-1.2.
pi@raspberrypi:~ $ 
```

Fig7. 9:Pip install face recognition-no-cache-dir

Face Recognition Project Folder

Our Project folder will consist of two python program called the Face_Trainner.py and Face_Recog.py. A folder called Face_Images which consist sample images of the persons who has to be recognized. A Classifier file called "haarcascade_frontalface_default.xml" which is used to detect faces. And finally a trainer file called "face-trainner.yml" which will be generated using the Face_Trainner.py program based on the Images present inside the Face_Images Folder. All the files in my project folder are shown below



CHAPTER 8

RESULTS AND DISCUSSIONS

Some of the results got by us are shown below

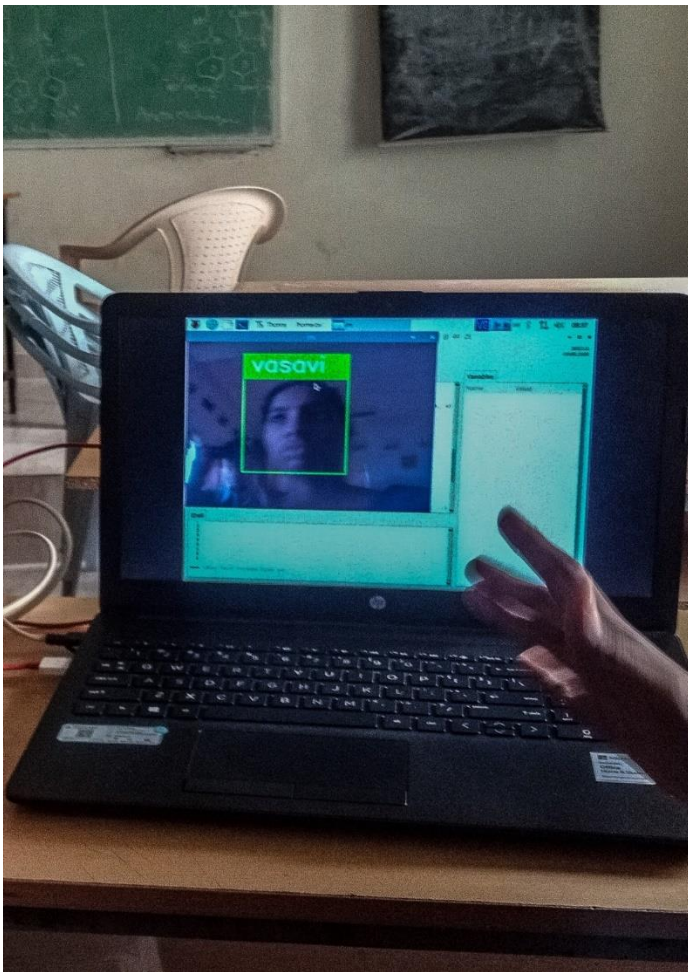


Fig8.1: User identification

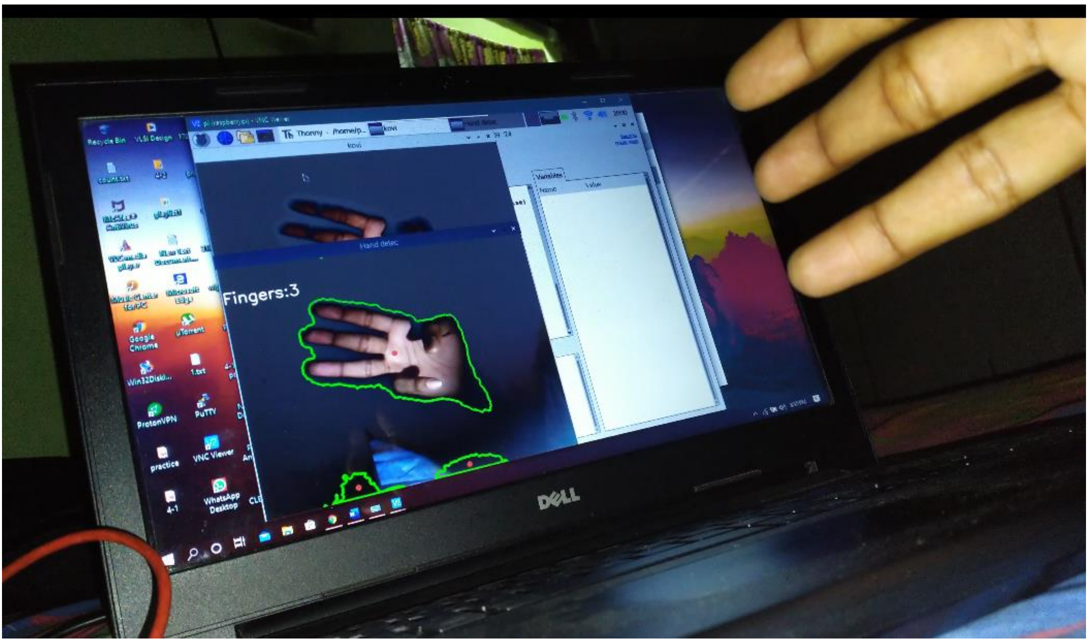


Fig8.2: Random finger number generation

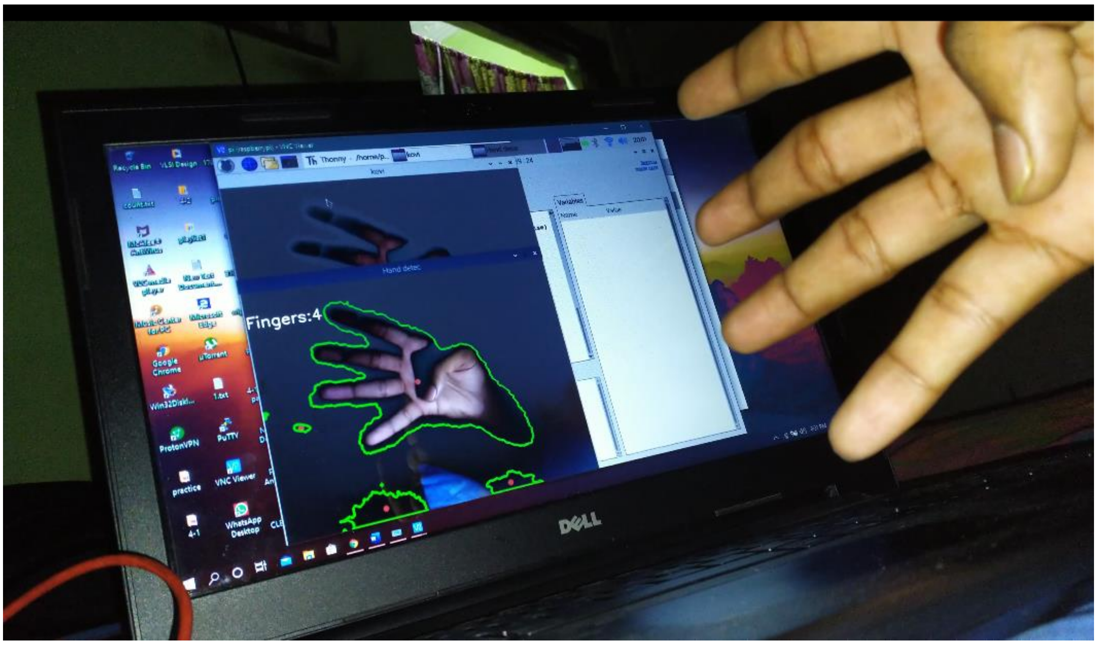


Fig8. 3: Random finger number identification

These pictures are the output when any particuar number is displayed then the user has to display same number in the screen we can see the number also. This is for additional security purpose.

CONCLUSION

In this proposed door access system by using face recognition the images are stored in a data base. This system uses door lock access for Residential and Commercial Purposes. Here we have designed a highly secured door locking system by using Raspberry pi.

Concern persons will be informed successfully about the person detection along with details attached. Face recognition is one of the several techniques for recognizing people. There are several methods that can be used for that purpose. Some of the most common are using PCA or eigenfaces. Though there are other new techniques more simple to understand the use and implement but also with very good performance. The Haar Cascades algorithm is one of those algorithms. As we show Haar Cascades has very good performance and is very accurate.

The complete door unlocking system is implemented in Open CV. This door access system shows the use of facial recognition techniques for the purpose of security.

REFERENCES:

1. D. Yang, Abeer Alsadoon, P.W.C. Prasad, A.K. Singh, A. Elchouemi, “An Emotion Recognition Model Based on Facial Recognition in Virtual Learning Environment”, Procedia Computer Science, Vol. 125, pp.2-10, 2018.
2. N. Kazanskiy, V. Protsenko, P. Serafimovich, “ Performance analysis of real-time face detection system based on stream data mining frameworks”, Procedia Engineering, Vol. 201, pp. 806-816, 20
3. Ratnawati Ibrahim, "Study of Automated Face Recognition System for Office Door Access Control Application", IEEE 2011
4. Fan X., Zhang F., Wang H., Lu X., “The System of Face Detection Based on OpenCV”, IEEE Control and Decision Conference (CCDC), 2012 24th Chinese,648-651, 2012.
5. R. Lienhart, J. Maydt, ”An extended set of Haar-like features for rapid object detection”, Vol. 46 Issue 3, pp. 518– 545. Dec. 2004.
6. Archana, T., Venugopal, T., & Kumar, M. P. (2015, January). Multiple face detection in color images. In Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on (pp. 82-85). IEEE.