IRIS BASED ATTENDANCE MANAGEMENT SYSTEM

Author

Capt Fawad Muzaffar

Capt M Akash Arshad

Capt M Abrar Nasir

(BETE 54A)

Supervisor

Dr M. Imran (Associate HOD)

Dr Ayesha Habib

Submitted to the faculty of Department of Electrical Engineering,

Military College of Signals, National University of Sciences and Technology,

in partial fulfillment for the requirements of B.E Degree in Electrical Engineering

(JUNE), 2021

CERTIFICATE OF CORRECTIONS & APPROVAL

Certified that work contained in this thesis titled*" IRIS BASED ATTENDANCE MANAGEMENT SYSTEM"* carried out by Fawad Muzaffar, M Akash Arshad & M Abrar Nasir under the supervision of Dr M. Imran Associate HOD and Dr Ayesha Habib for partial fulfillment of Degree of Bachelors of Electrical Engineering, in Military College of Signals, National University of Sciences and Technology, Islamabad during the academic year 2020-2021 is correct and approved. The material that has been used from other sources it has been properly acknowledged / referred.

Approved by

Supervisor

Dr M. Imran

Assoc Prof

Date: __________________

DECLARATION

No portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

Capt Fawad Muzaffar

Capt M Akash Arshad

Capt M Abrar Nasir

Signature of Supervisor

Dr M Imran
Assoc Prof

Acknowledgements

I am thankful to Almighty to have guided me throughout this work at every step and for every new thought which You setup in my mind to improve it. Indeed I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life.

I would also like to express special thanks to my supervisor Dr M .Imran Associate HOD and Dr Ayesha Habib for their help throughout my thesis .I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.

Abstract

Manual attendance system is one of time-consuming process and the human error remains. There are many methods available for attendance system like fingerprint and face recognition. The most effective method of attendance system is based facial recognition and iris-based attendance system. We have developed an iris-based attendance system in which the live camera takes the picture of user eye, recognize it and mark its attendance and save it in CSV file. Our project consists of automatic mail sending, which send the email while click the email tab when attendance is done. The whole system of this project is based on Raspberry pi based single board computer and raspberry pi camera.

Table of content

List of Figures

Chapter 1

1. Introduction

1.1 Background

Many attendance systems are available, RFID based attendance system was the most prominent and old system which were used for attendance and also used for security system. The drawback of RFID based attendance system is that it can be lost and also mis used. This system is not secure and reliable.

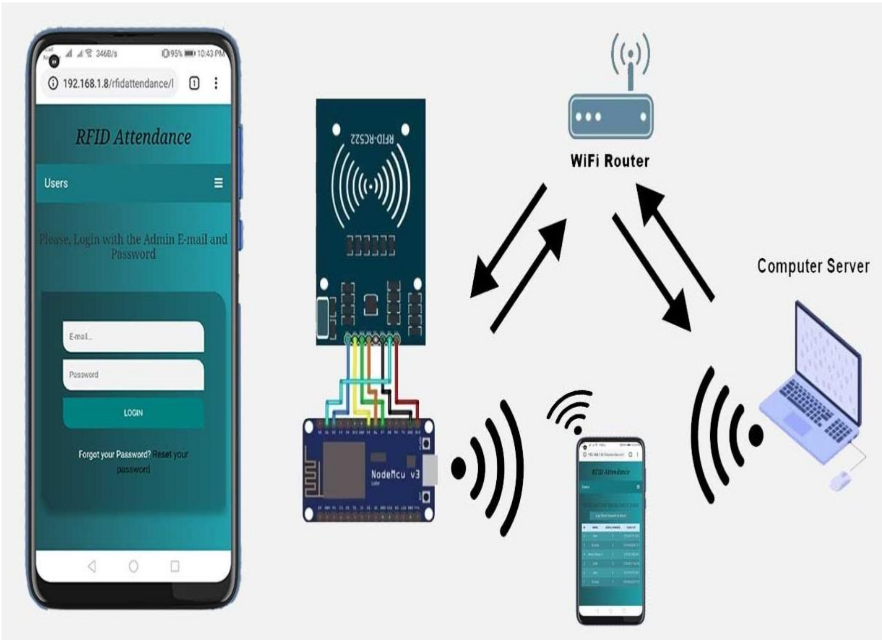


Figure 1 RFID based Attendance system

Biometric attendance system is also well known and prominent system in which fingerprints are used, but the drawback of this attendance system is that it can not be used for those who are disable like the peoples have no arms, no hand or having no fingers. This system also has limitations. This system is dedicated but there is difficult process to enroll and delete the peoples in this attendance system.

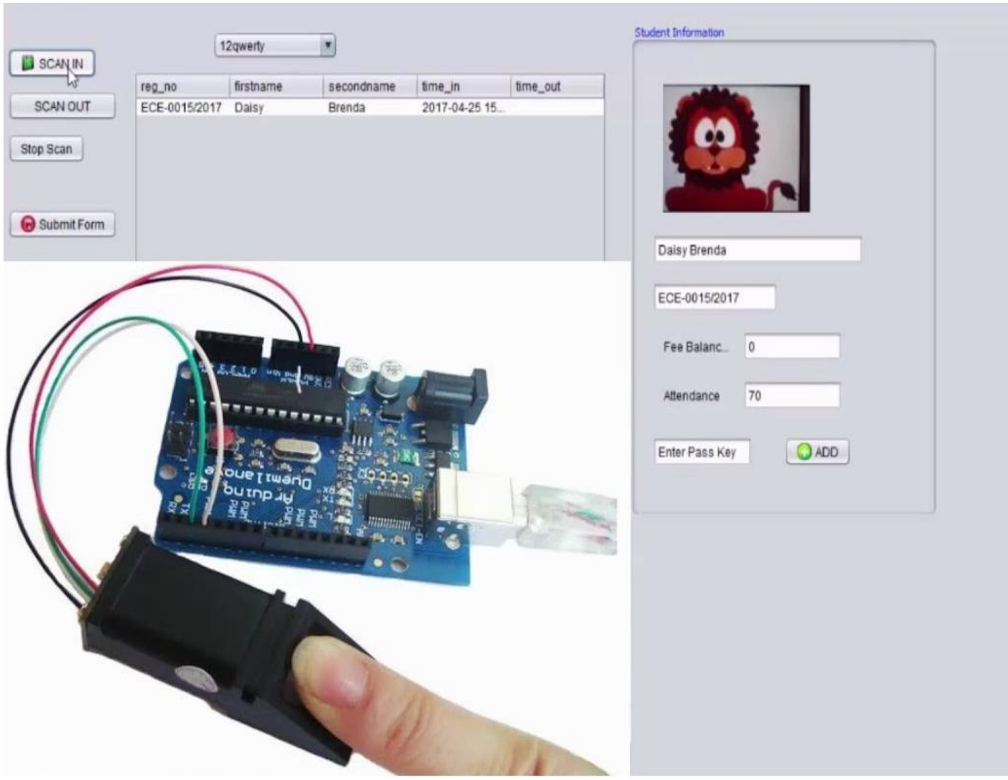


Figure 2 Fingerprint based attendance system

The new method of attendance system is based on IRIS, this method is more accurate and has a very less limitations, the only limitation in this system is that a person has no eyes and, very small number of peoples are those have no eyes. In iris-based attendance system, there are many algorithms available to recognize the IRIS, but the most easy and useful algorithm is voila jones which results are less accurate but it can run on any Raspberry pi-based board.

The most prominent algorithm is tensor flow which is more advance and accurate. This algorithm required much computational power and heavy processor. We have used Raspberry pi-based operating system which has a processor of 1.2GHz and the RAM of 1GB. In which the tensor flow can run around 1 FPS, or stream the video on 1 or 1.5 frame per second.

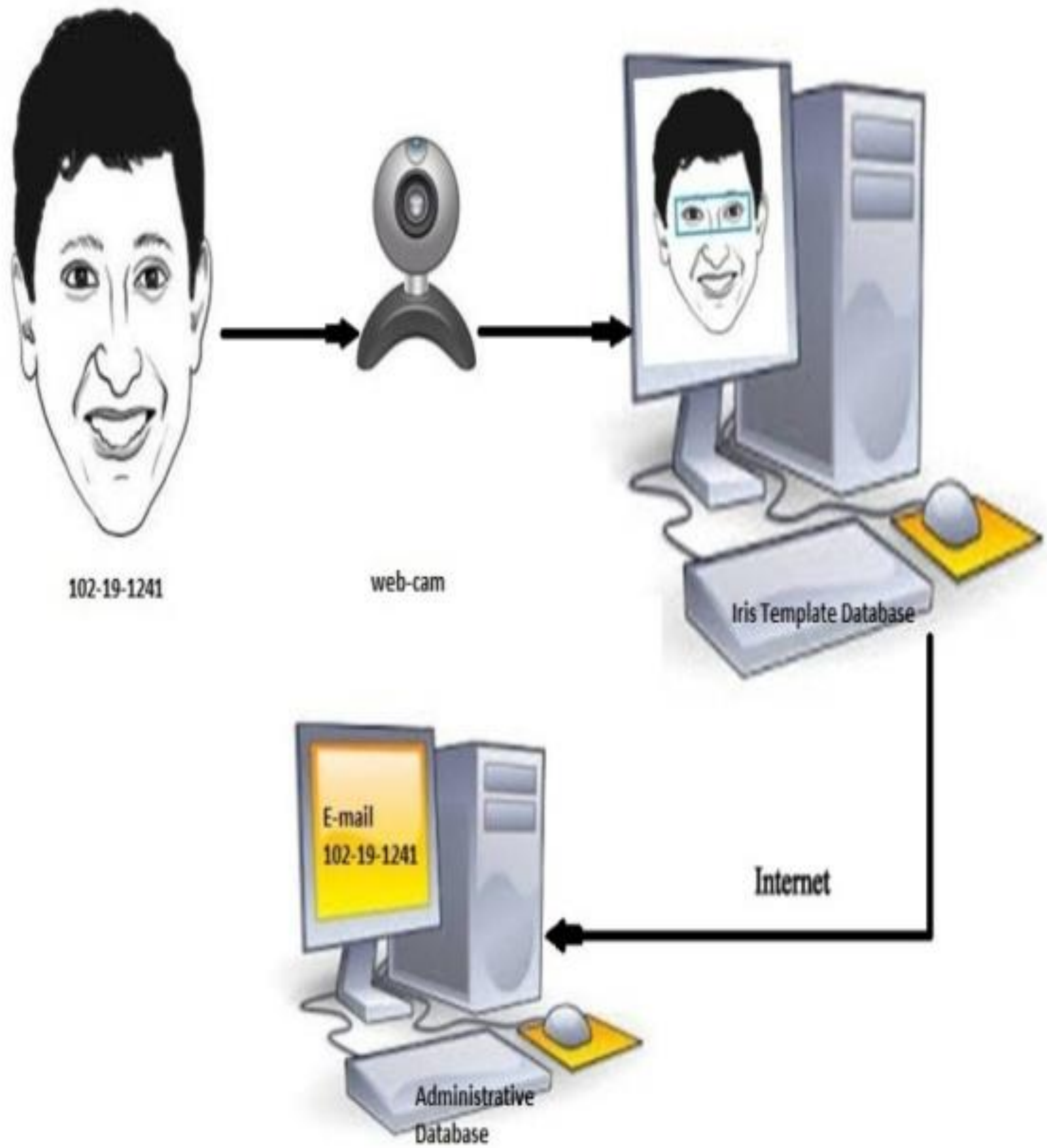


Figure 3 IRIS based attendance system with data base in computer

The figure explains the IRIS based attendance system in which the complete computer system is involved and it is not portable. We required a portable attendance system so that we consider the raspberry pi and Raspberry pi camera.

1.2 Problem: IRIS based attendance system

Need of efficient accurate, and error free attendance system in which there is less time consuming and accurate attendance system. IRIS and face based attendance system was proposed and implementation of eye retina based attendance system is designed.

1.3 Thesis Statement

The basic idea of this thesis is the attendance system based on IRIS along with automatic email sending. The system is based on Raspberry pi based single board computer and pi camera which require less power and is of portable which can operate on 5 V and it can also operate on battery or power bank.

1.4 Approach

In the process of designing attendance system, we have selected the hardware and its accessories for image processing, In image processing we have installed OpenCV library for algorithms implementation.

1.5 Potential Impact

This attendance system creates a major impact on reliable attendance system in which there is less chance of human error. This system is low cost, low power and portable.

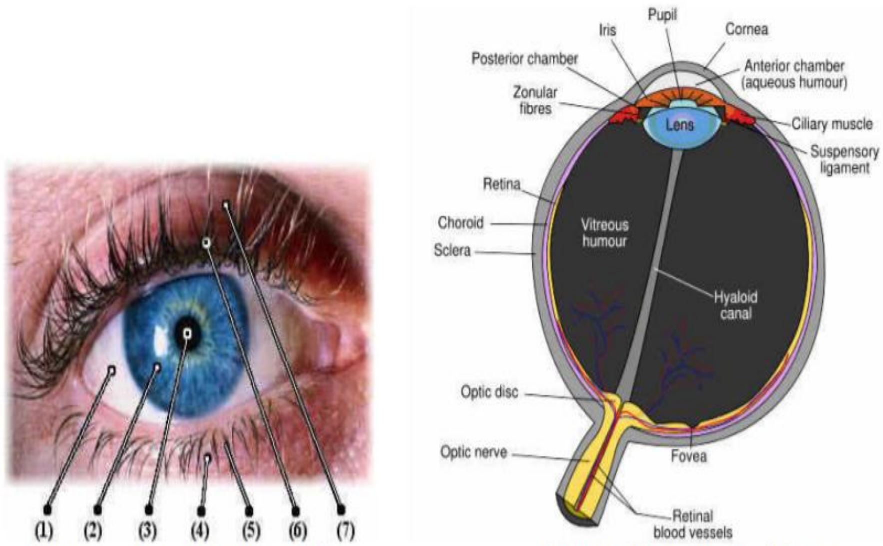
1.6 Organization

This thesis has been organized in five chapters. First chapter is the introductory chapter as introduces the topic of the thesis and includes the main idea behind the research. Second Chapter gives a brief overview of the literature review performed in order to point out the previous work on this topic The third chapter is based on the detailed study of proposed algorithm and hardware. Fourth chapter of the thesis concludes our thesis with the analysis of the system and the conclusions of the experiments.

Chapter 2

2. Literature Review

In manual attendance system there are many problems like, human error, proxy based attendance. The manual attendance system effects the industry outcomes., because of the laziness of employee and time commitment issue. Iris recognition is one of the most authentic, reliable and accurate method for attendance system.



Human eye, where (1) is sclera, (2) iris, (3) pupil, (5, 6) eyelashes And (4,7) eyelids.

Anatomy of frontal image of the eye.

Figure 4 detailed Structure of Eye

The above figure explains the human eye classification in which seven parts has been highlighted [1]. In this paper the author used the approach of feature extraction. The process includes taking image, feature extraction and classification.

Project	Country	Enrollments	Purpose
UIDAI Aadhaar	*India*	*>200 million (2012)*	*National identity number for financial aids, governance, etc. [20]*
UNHCR Repatriation	*Afghanistan*	*>2 million (2008)*	*Refugee registration for aid (assistance package, food, etc.) [21]*
UAE Border control	*UAE*	*>1.5 million (2008)*	*Expellees tracking and border control in the United Arab Emirates [18]*
TSA CLEAR	*US*	*>175 thousand (2008)*	*Frequent traveler border crossing at major US airports [17]*
National Airport Sec.	*Canada*	*>150 thousand (2010)*	*Security solution at 29 Canadian airports [16]*
IRIS	*UK*	*>100 thousand (2009)*	*Heathrow, Manchester, Birmingham and Gatwick airports border control [14]*
NEXUS	*US, Canada*	*>100 thousand (2006)*	*Frequent traveler border-crossing [22]*
York County Prison	*US*	*>35 thousand (2012)*	*Inmates registration [16]*
Privium	*Netherlands*	*>30 thousand (2009)*	*Frequent traveler border control at Schiphol airport using smartcard [15]*

The above table explains the history of public deployment project of iris recognition system.

The author implements the iris recognition using MATLAB by taking input image, train them and then test the code using real image. In this paper no webcam is used for real time image.[2]. The process is to take the image, extract features, train the data and test the final code.

In this paper the author used both the approaches face and iris-based recognition system in which face is recognized and iris also used for recognition. The two steps have been used for final decision. First the system take images of face and eye, train them and generate one training file from the images, the training file generated in the form of .yml. After training the testing data is used [3].

The author used RCNN based iris recognition system in which the author first develop the algorithm and take the pretrained data set from website and used that data to recognize the iris. After that the author develop the customized data set and test the real time code using images. The limitation if this developed system is that a high speed and accurate pixel camera is required [4].

The author develop the iris recognition system using MATLAB and test the result, the limitation of this system is very slow and its very difficult to implement the system in real time, because MATLAB create a lot of hardware delay [6].

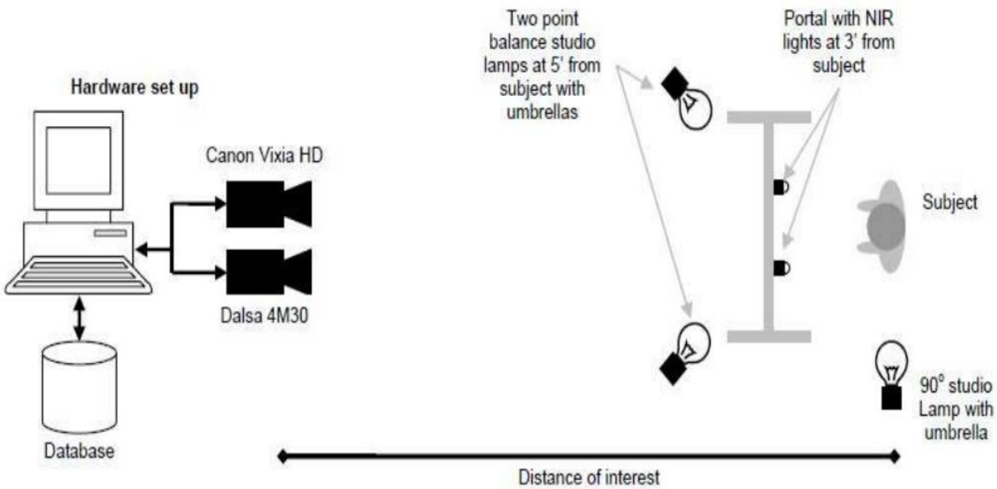


Figure 5 Experimental setup used by author

In the above figure the author arrange a complete setup of iris recognition system in which two camera's are used and NIR lights are also used[9].

The author used Fast RCNN based iris recognition system in which the author first develop the algorithm and take the pretrained data set from website and used that data to recognize the iris. After that the author develop the customized data set and test the real time code using images. The limitation if this developed system is that a high speed and accurate pixel camera is required [5].

Most of the iris acquisition system are based on NIR based this image acquired using LG2200 sensor which is near infrared.

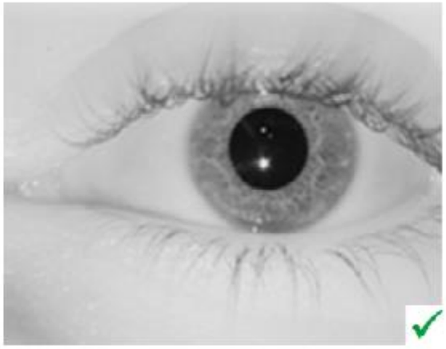


Figure 6 good quality EYE image

This image is successfully recognized using near infrared sensor.

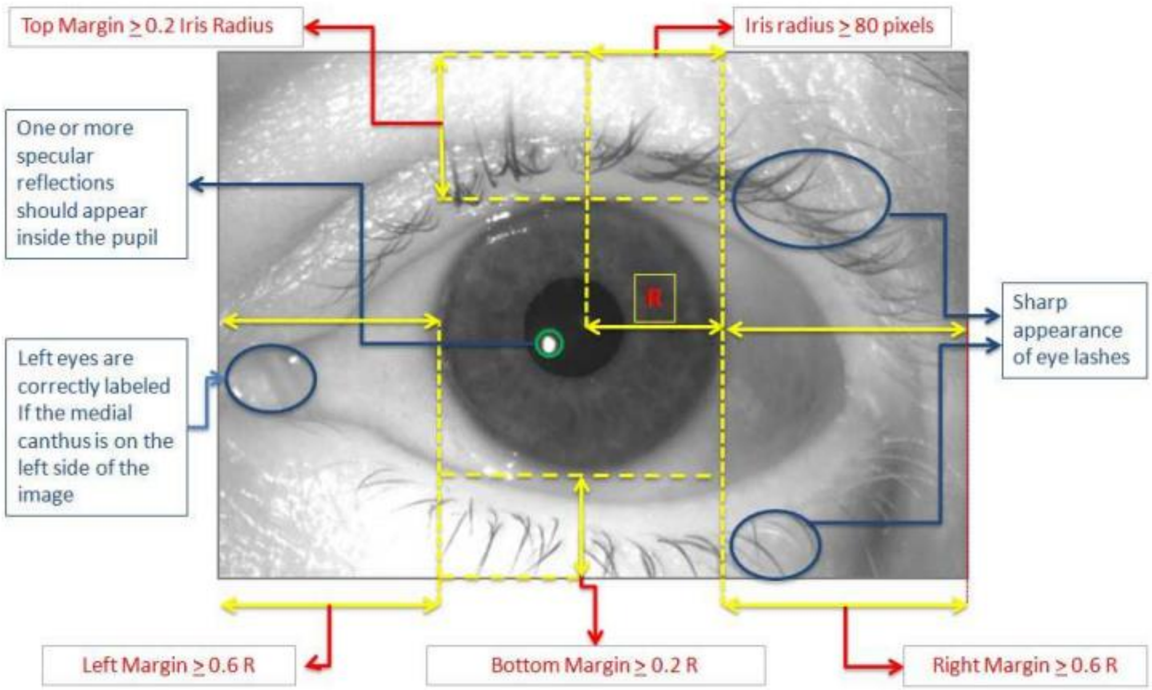


Figure 7 Complete Internal Structure of IRIS

The correct Iris recognition image is shown above in which each segment is clearly defined.

Chapter 3

3. Implementation

In 1993 Doughman proposed the complete iris recognition system first time, and most of the iris recognition systems are working on same principle. We have equation (1) as following.

$$\max_{r,x_0,y_0} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right|$$

In above equation (1) represent the original iris image, G represents the gaussian filter and sigma represents the standard deviation. We have equation (ii) as,

$$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta)$$

Where r represents the range from [0,1] and theta represent the angle between [0,2pi], where y represents the combination of pupil's points.

$$x(r,\theta) = (1-r)x_p(\theta) + rx_i(\theta)$$

$$y(r,\theta) = (1-r)y_p(\theta) + ry_i(\theta)$$

The normalized iris image has many benefits like it can be easily compared.

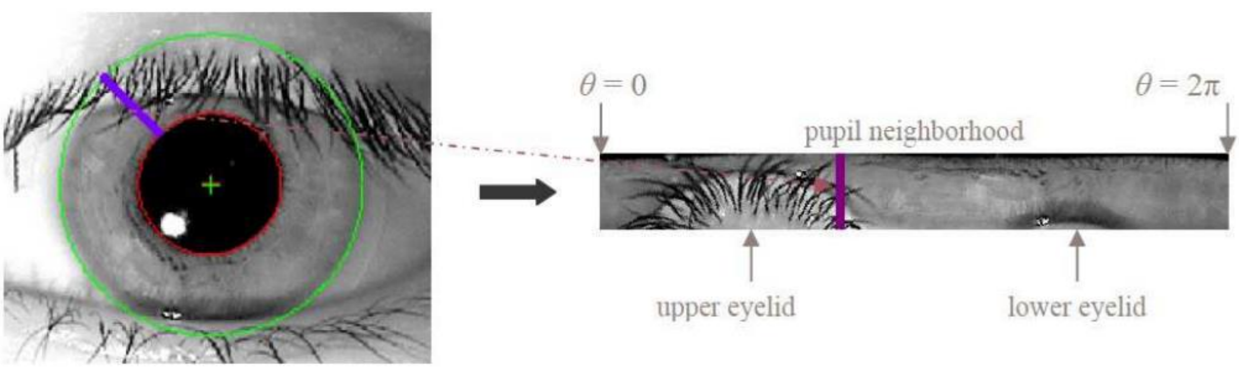


Figure 8 Iris Image detail with Mathematics

The above image shows the detail of iris recognition system in which complete detail with parameters are shown.

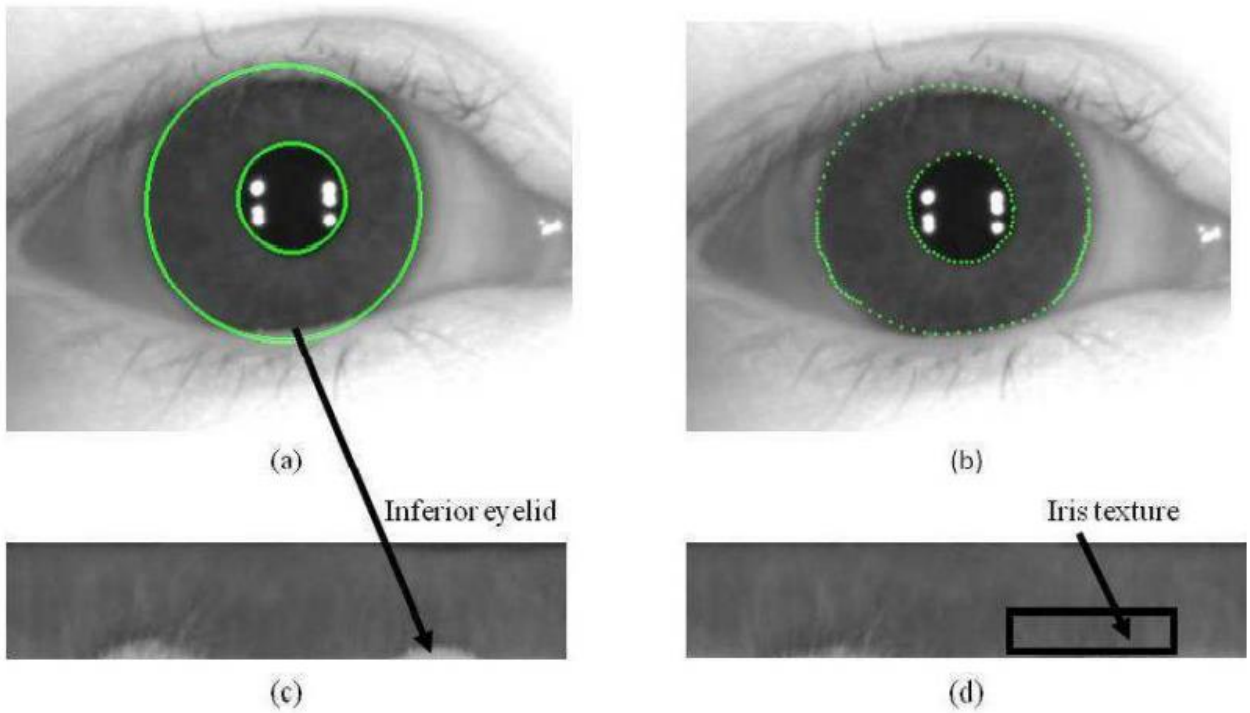


Figure 9 Detailed Eye structure

In iris recognition system there is a big concern of camera image quality and also the image quality of iris image, the feature extraction required high resolution image.

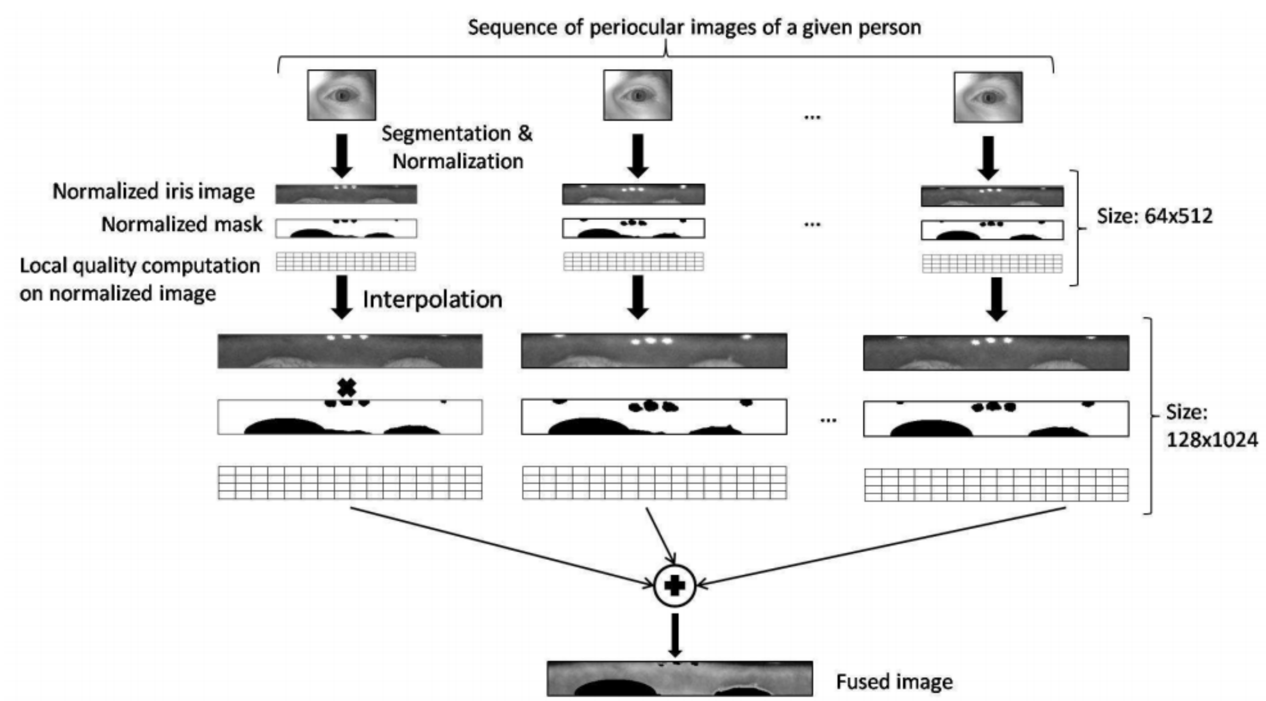


Figure 10 iris recognition process step by step

The above picture shows the fusion process of local quality-based images. There are three steps of this method in which first we take the periocular image of a person iris. The second step is to normalize the iris image. The last step is to quality computation on normalized image.

$$I(x, y)_{fused} = \frac{\sum_{i=1}^{F} I^i(x, y) \cdot M^i(x, y) \cdot Q^i}{\sum_{i=1}^{F} M^i(x, y) \cdot Q^i}$$

In fusion process, the reconstructed image is obtained by using the above equation.

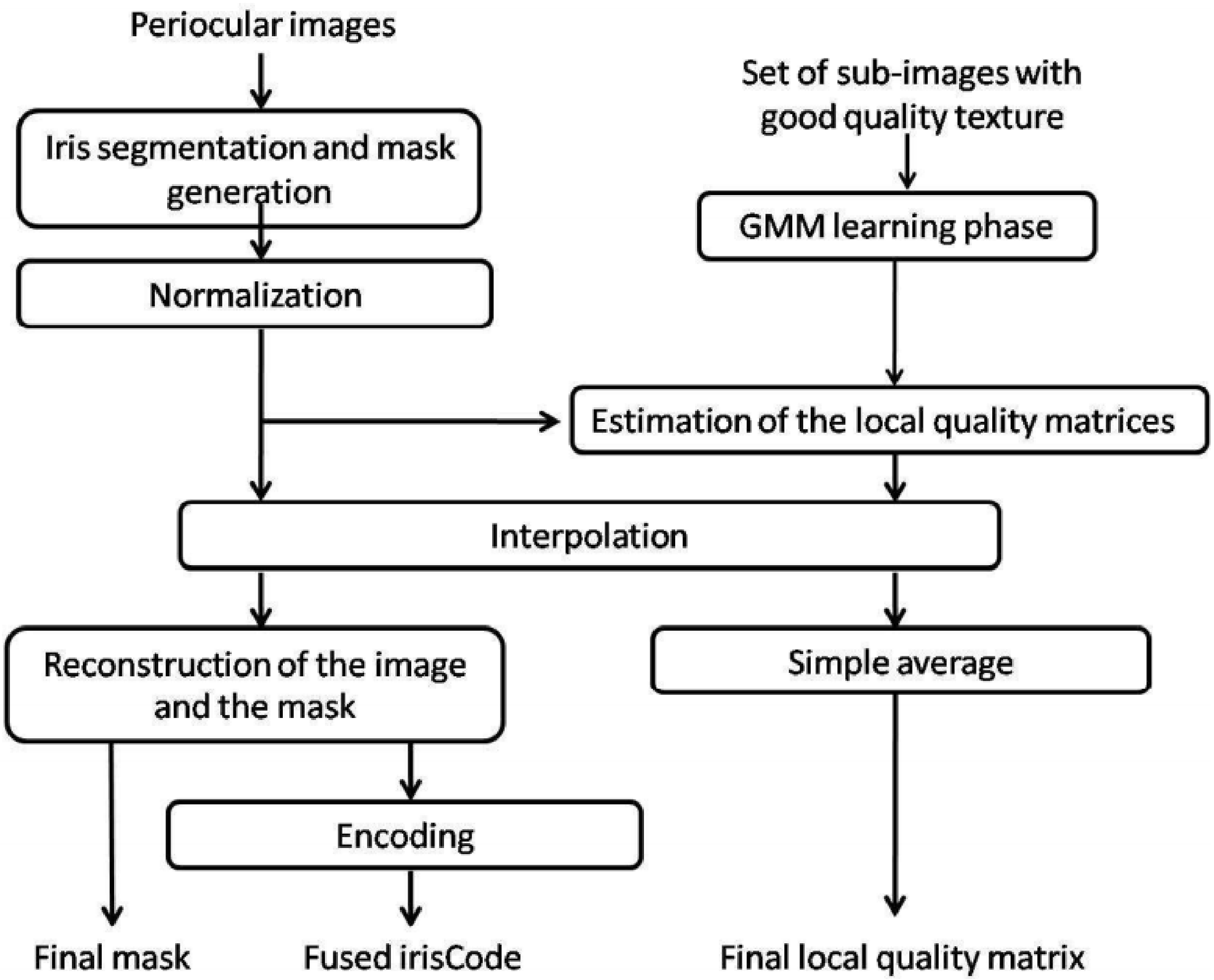


Figure 11 Flow diagram of IRIS recognition

3.1 Iris Recognition Process

The IRIS recognition consists of following major parts which are as follows.

- Iris Segmentation
- Normalization
- Feature extraction
- Template Matching

3.1.1 Iris Segmentation

The image has extra information available; we need to remove the noise. The image segmentation process convert the image into binary to indicate that which segment belong to iris texture and which belong to extra noise.

3.1.2 Normalization

In normalization process the iris image is mapped into size invariant band called normalized iris image.

3.1.3 Feature Extraction

In this process the aim is to extract the texture characteristic of given iris image.

3.1.4 Template Matching

It is the final stage, this system decides whether the template belong to same iris or not. At this end the matching and dissimilarity score is calculated.

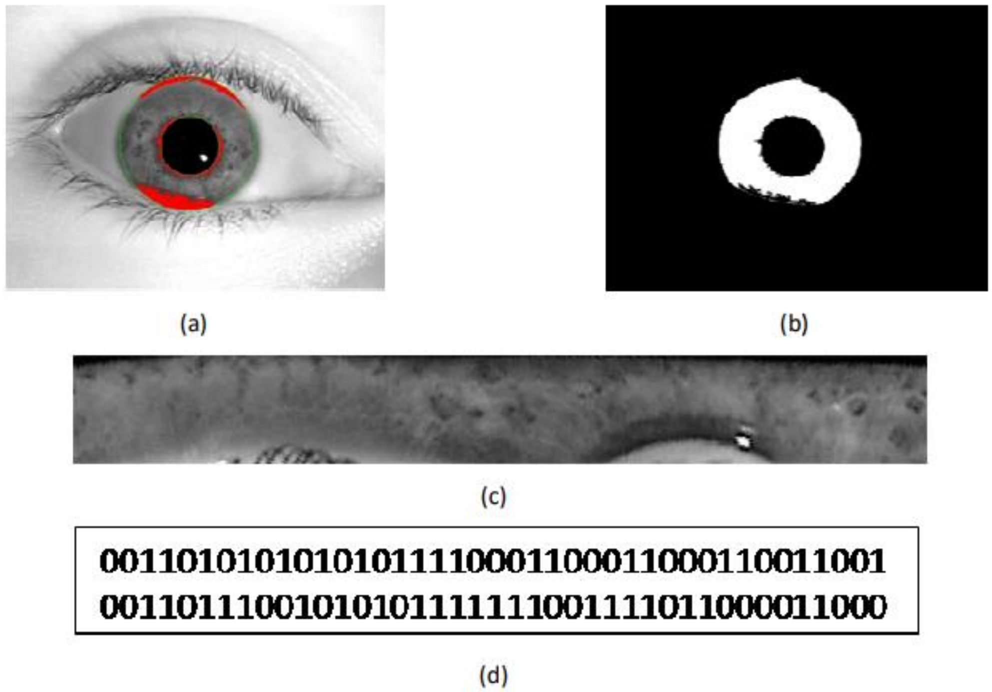


Figure 12 Iris Image processing detail

The above figure shows the four stages of image, first one shows the image segmentation, the second one shows the iris mask, the third one image shows the normalized iris image and the last one shows the binary image values.

3.2 Hardware Design

We have developed a portable iris recognition system in which the system hardware components are required which are of low power and can easily be operate on 12V battery or power bank.

The components required for the development of hardware is are as follows

- Raspberry Pi 3 B+
- Raspberry Pi Camera
- DC gear motor
- Motor driver circuit
- Power Bank
- LCD

- Keyboard.

3.2.1 Raspberry Pi 3 B+

We have used Raspberry pi 3 B+ as the main processing unit which can easily operate on 5V power bank and consume less power. Although it has multiple variants available but Raspberry Pi 3 B+ is compatible to our purpose offering 1.2 GHz quadcore processor, built in Wi-Fi, Bluetooth, audio jack and IO lines for different sensors interfacing. Furthermore its camera slot interfaces pi camera with raspberry. Webcam can also be interfaced with board through any of the USB port available .

Figure 13 Raspberry Pi 3 B+

Raspberry Pi 3 B+ in above diagram has built in full HDNI port for display, along with all other built in accessories similar to a single computer board. Moreover we can install any software in it by using terminal (command prompt).

3.2.2 Key Board

Keyboard basically serve as main part of raspberry pi to select the program, to display the figures and videos.

Figure 14 Keyboard used for Raspberry Pi

So prefer normal keyboard for proving input to the software.

3.2.3 LCD Screen

Utility of LCD screen is for the display of information and to give certain commands to raspberry pi, Here we preferred HDMI based LCD by utilizing built in HDMI port of raspberry pi.

Figure 15 HDMI LCD SCREEN for Raspberry Pi

This screen is based on HDMI and is used to display the complete information of raspberry pi.

3.2.4 DC gear motor

We have designed one barrier which consist of one DC gear motor for opening and closing the barrier. This system. The dc gear motor is of 12V and 200mA. Which is controlled by Raspberry pi.

Figure 16 DC Gear Motor

3.2.5 DC gear motor interface with Raspberry pi

For barrier control we have used dc gear motor and the dc gear motor is controlled by raspberry pi in forward as well as reverse direction.

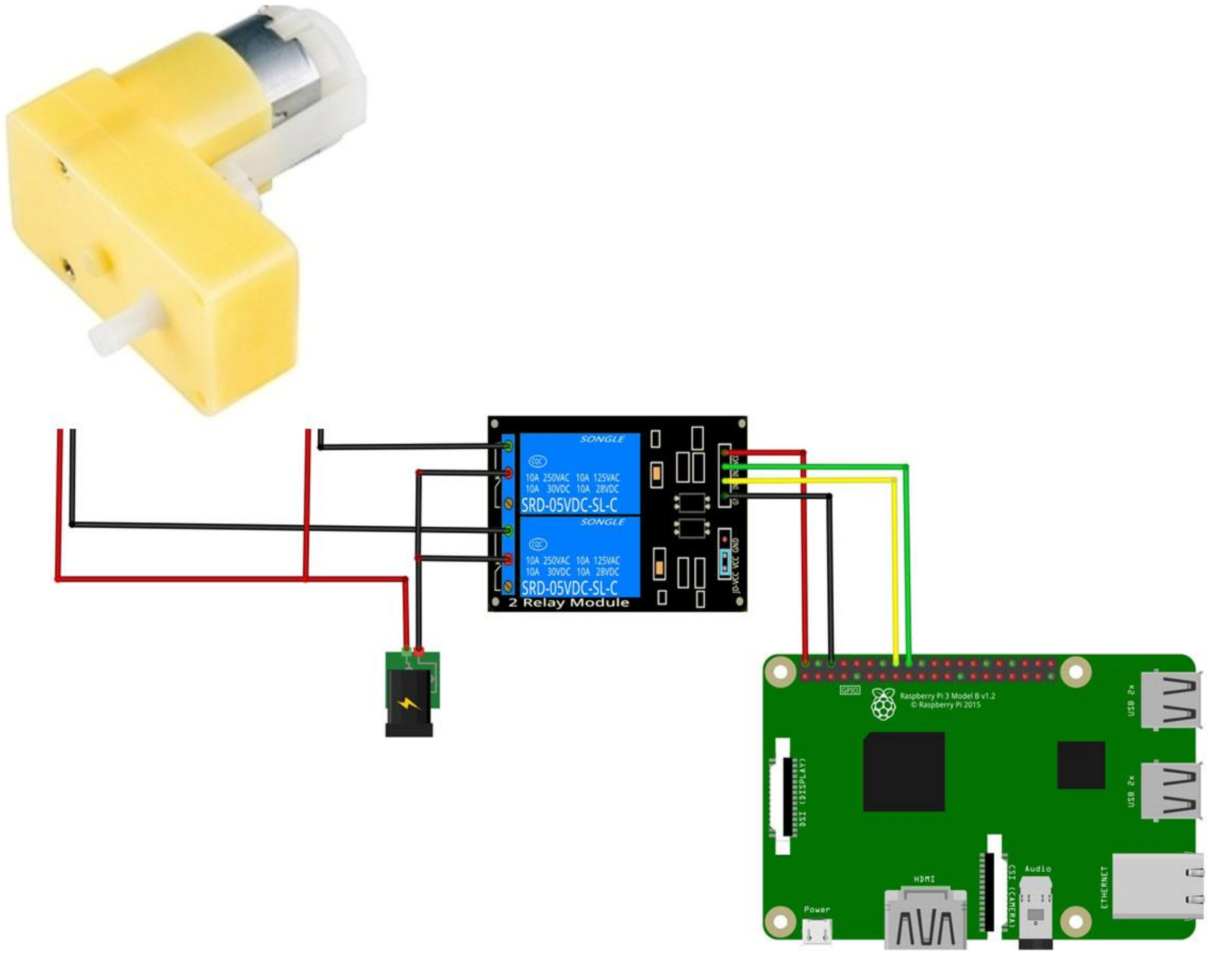


Figure 17 Interfacing of DC gear motor With Raspberry pi

3.2.6 Raspberry pi

We have used raspberry pi camera of 5MP which is used for taking real time images.

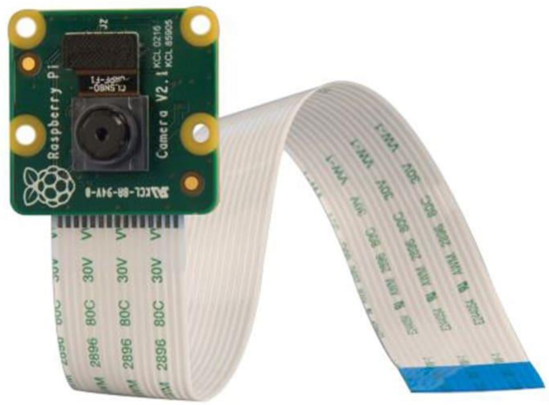


Figure 18 Raspberry Pi Camera

Raspberry pi has its own camera which can be directly interfaced with raspberry pi.

3.3 Image processing

For image processing we have used open CV library which is a very power full library used for the execution of image processing codes.

Before installation of OpenCV we prepare the system using following command

```
sudo apt-get -y purge wolfram-engine
sudo apt-get -y purge libreoffice*
sudo apt-get -y clean
sudo apt-get -y autoremove
```

First of all, we clean the build directory using following command in these command we use sudo which is used to run the command as an administrator

```
# Clean build directories
rm -rf opencv/build
rm -rf opencv_contrib/build
```

After cleaning the build directory, we make new installation directory

```
mkdir installation
mkdir installation/OpenCV-"$cvVersion"
```

For updating the package, we use following commands

```
sudo apt -y update
sudo apt -y upgrade
```

Before installation of open CV some libraries are required to install which are pre requisite of this library

```
sudo apt-get -y remove x264 libx264-dev

## Install dependencies
sudo apt-get -y install build-essential checkinstall cmake pkg-config yasm
sudo apt-get -y install git gfortran
sudo apt-get -y install libjpeg8-dev libjasper-dev libpng12-dev

sudo apt-get -y install libtiff5-dev

sudo apt-get -y install libtiff-dev

sudo apt-get -y install libavcodec-dev libavformat-dev libswscale-dev
libdc1394-22-dev
sudo apt-get -y install libxine2-dev libv4l-dev
cd /usr/include/linux
sudo ln -s -f ../libv4l1-videodev.h videodev.h
cd $cwd
```

The above libraries include the images, videos and codec libraries

```
sudo apt-get -y install libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev
sudo apt-get -y install libgtk2.0-dev libtbb-dev qt5-default
sudo apt-get -y install libatlas-base-dev
sudo apt-get -y install libmp3lame-dev libtheora-dev
sudo apt-get -y install libvorbis-dev libxvidcore-dev libx264-dev
sudo apt-get -y install libopencore-amrnb-dev libopencore-amrwb-dev
sudo apt-get -y install libavresample-dev
sudo apt-get -y install x264 v4l-utils
```

These libraries and plugins are required for the installation of OpenCV.

```
sudo apt-get -y install python3-dev python3-pip
sudo -H pip3 install -U pip numpy
sudo apt-get -y install python3-testresources
```

In raspberry there are built in installed software are python 2 and python 3

NumPy library is required for image processing

```
cd $cwd
# Install virtual environment
python3 -m venv OpenCV-"$cvVersion"-py3
echo "# Virtual Environment Wrapper" >> ~/.bashrc
echo "alias workoncv-$cvVersion=\"source $cwd/OpenCV-$cvVersion-
py3/bin/activate\"" >> ~/.bashrc
source "$cwd"/OpenCV-"$cvVersion"-py3/bin/activate
#############
```

The above commands are used for creating virtual environment

```
git clone https://github.com/opencv/opencv.git
cd opencv
git checkout $cvVersion
cd ..

git clone https://github.com/opencv/opencv_contrib.git
cd opencv_contrib
git checkout $cvVersion
cd ..
```

These commands are used for downloading the opencv from github using git command

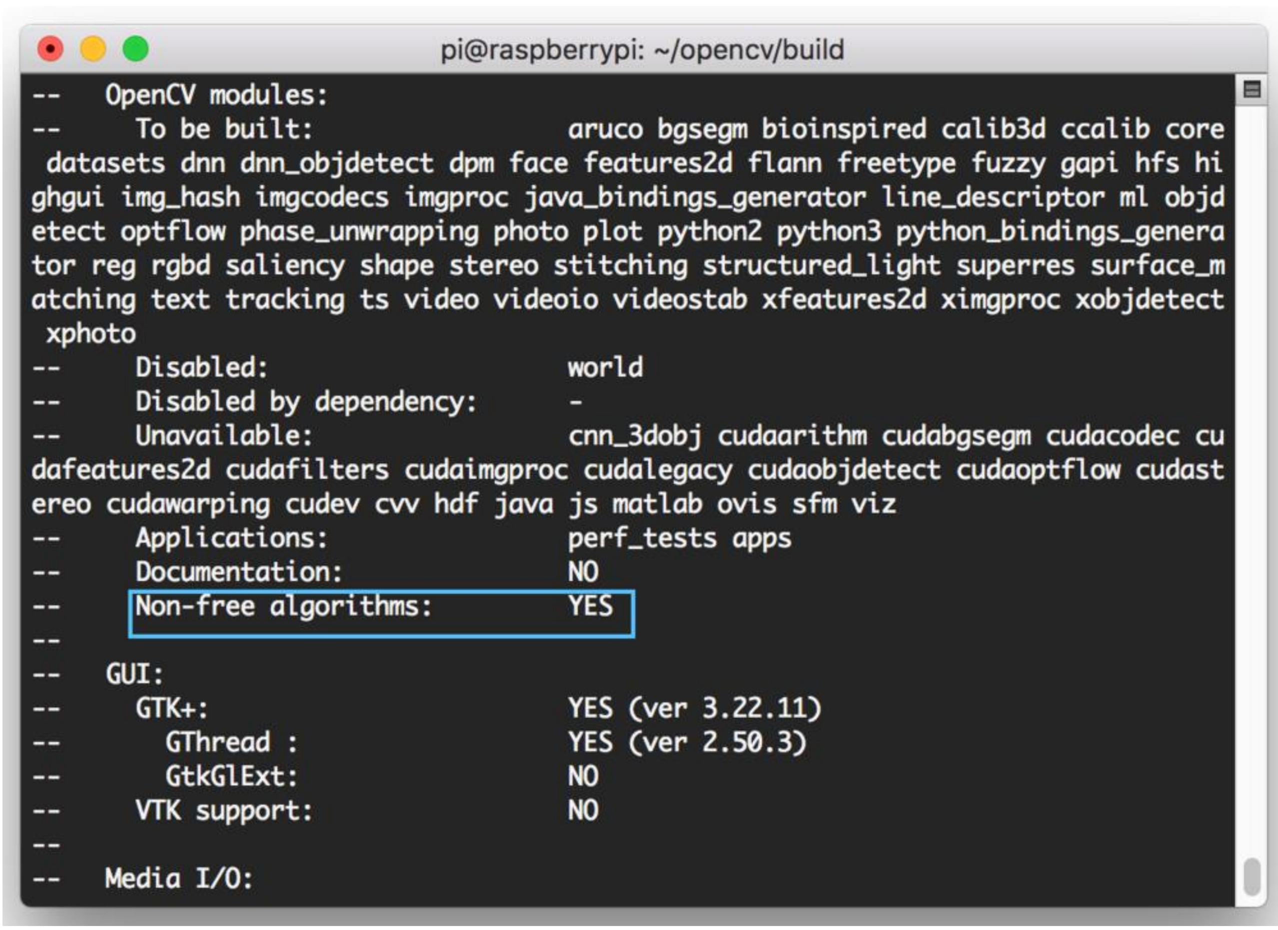


Figure 19 OpenCV installation Process

```
cd opencv
mkdir build
cd build
```

Here we create the build directory for OpenCV installation

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
            -D CMAKE_INSTALL_PREFIX=$cwd/installation/OpenCV-"$cvVersion" \
            -D INSTALL_C_EXAMPLES=ON \
            -D INSTALL_PYTHON_EXAMPLES=ON \
            -D WITH_TBB=ON \
            -D WITH_V4L=ON \
            -D OPENCV_PYTHON3_INSTALL_PATH=$cwd/OpenCV-$cvVersion-
py3/lib/python3.5/site-packages \
            -D WITH_QT=ON \
            -D WITH_OPENGL=ON \
            -D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib/modules \
            -D BUILD_EXAMPLES=ON ..
```

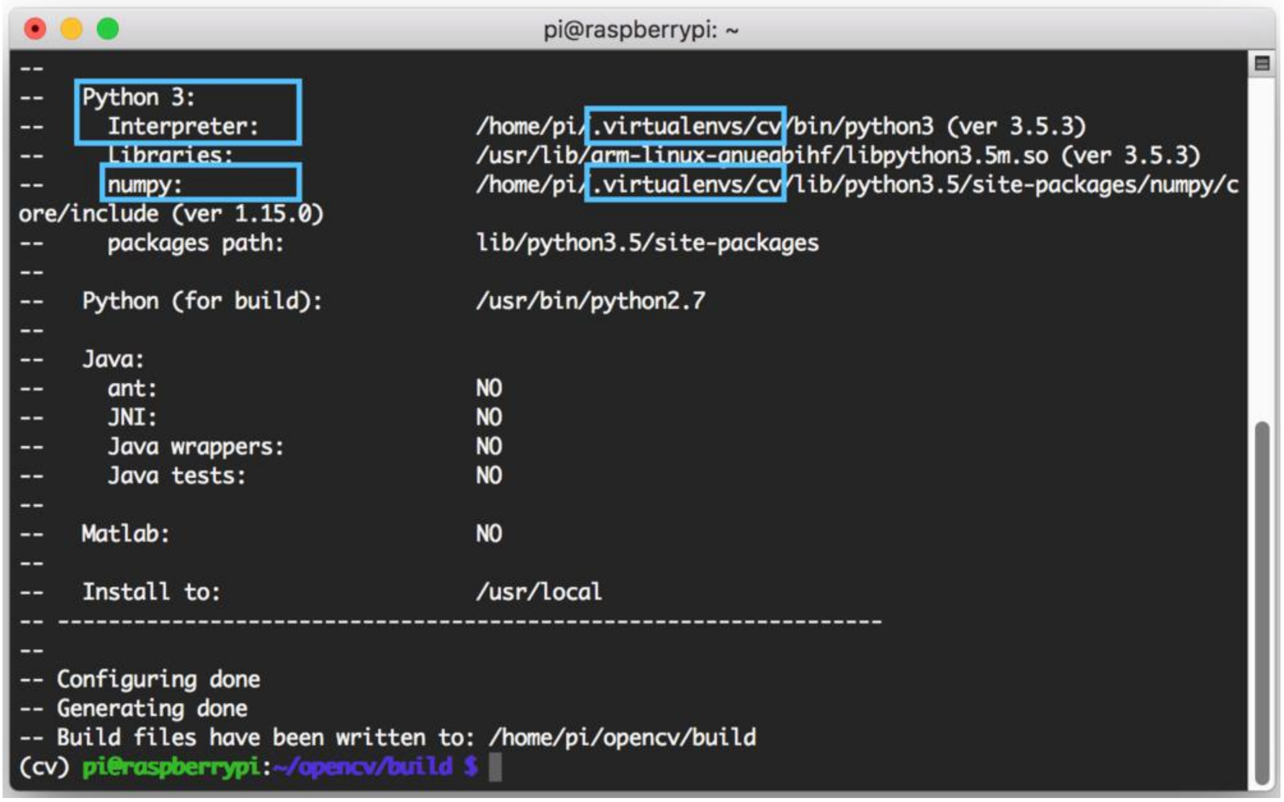


Figure 20 Build Open CV process

Here we check the status of installed libraries

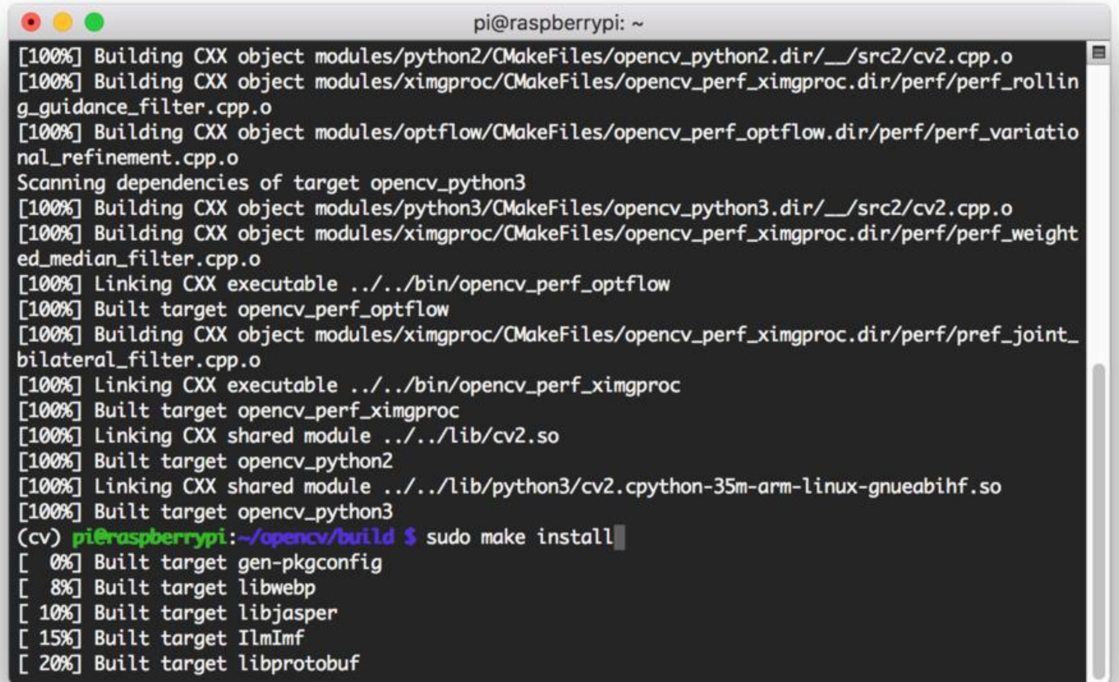


Figure 21 Dependencies Installation Process

Make is used to compile the open CV library and after compilation we installed it in Raspberry pi.

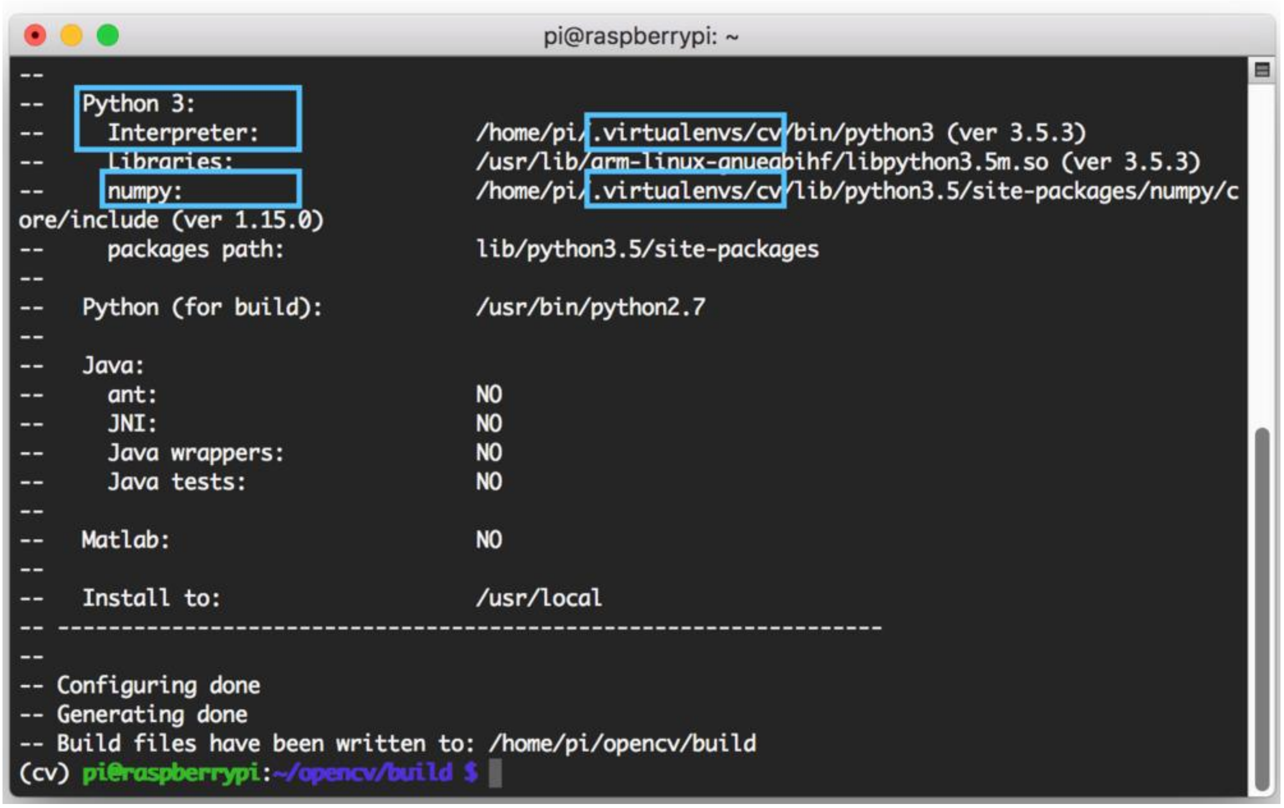


Figure 22 Testing open CV

3.4 Image Processing detail

We have used OpenCV library for image processing, this library can be used for both python and C++. We have installed the dependencies required for image processing.

```
import numpy as np
```

In python we used import for including the libraries, NumPy is a library used for image processing.

```
import cv2
```

Cv2 is an open CV library used for image processing

We have used voila jones algorithm for iris recognition, voila jones is an algorithm used for vast recognition-based applications.

```
eye_cascade = cv2.CascadeClassifier('haarcascade_righteye_2splits.xml')
```

The voila jones algorithm is also called Haar cascade in which many trained data sets are available and is of light weight which can easily run-on raspberry pi. In above line we include the classifier for eyes ratina.

```
cap = cv2.VideoCapture(0)
```

Most of the peoples done their work on image based, because for iris recognition we need a high-end camera. Here we for video streaming we used normal raspberry pi camera, and initialize it using cv2.videocapture.

For forever loop we used while (1), which run the program infinite times.

```
ret, img = cap.read()
```

Here we read the image from camera.

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

We need to convert the color image to gray scale for the minimization of processing. Gray scale image required less pixels than the 24-bit color image.

```
eyes = eye_cascade.detectMultiScale(gray)
```

Here we create the object of eyes which used to detect eye from gray scale image.

```
for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(img,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
        roi_gray2 = gray[ey:ey+eh, ex:ex+ew]
        roi_color2 = img[ey:ey+eh, ex:ex+ew]
        circles = cv2.HoughCircles(roi_gray2,cv2.HOUGH_GRADIENT,1,20,param1=50,param2=30,minRadius=0,maxRadius=0)
        try:
            for i in circles[0,:]:
                # draw the outer circle
                cv2.circle(roi_color2,(i[0],i[1]),i[2],(255,255,255),2)
                print("drawing circle")
                # draw the center of the circle
                cv2.circle(roi_color2,(i[0],i[1]),2,(255,255,255),3)
        except Exception as e:
            print e
```

Here we detect eye retina and place circles across retina.

```
cv2.imshow('img',img)
k = cv2.waitKey(30) & 0xff
```

Here we show the preview video.

```
cap.release()
cv2.destroyAllWindows()
```

Here the above code destroys all windows.

```
for contour in contours:
      area = cv2.contourArea(contour)
      rect = cv2.boundingRect(contour)
      x, y, w, h = rect
      radius = 0.15 * (w + h)

      area_condition = (100 <= area <= 200)
      symmetry_condition = (abs(1 - float(w)/float(h)) <= 0.2)
      fill_condition = (abs(1 - (area / (math.pi * math.pow(radius, 2.0)))) <= 0.4)
      cv2.circle(frame, (int(x_eye + x + radius), int(y_eye + y + radius)), int(1.3 * radius), (0, 180, 0), -1)
```

The above code is used to detect contour of eye image and circles shown on pupils.

```
cv2.imshow('Pupil Detector', frame)
c = cv2.waitKey(1)
```

The above code shows the pupil detector display.

Figure 23 Test Image of IRIS detection

Chapter 4

4.1 Results and Discussion

The results of this project is obtained from different algorithms like first we use canny edge detection, the canny edge detection has 5 steps the steps are as follows

Figure 24 IRIS input Image

The very first step is to read the input image.

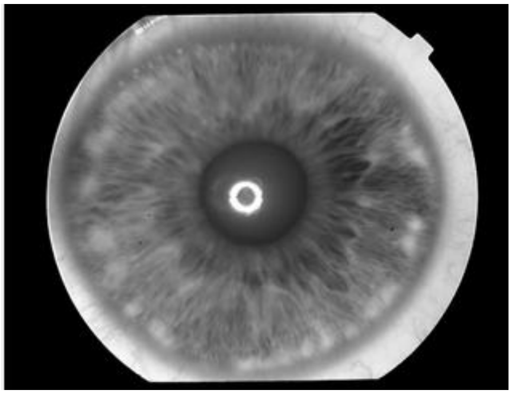


Figure 25 Gray Scale Image

The second step is to convert the input image to gray scale image. This gray scale image decreases the weight of image with almost same information.

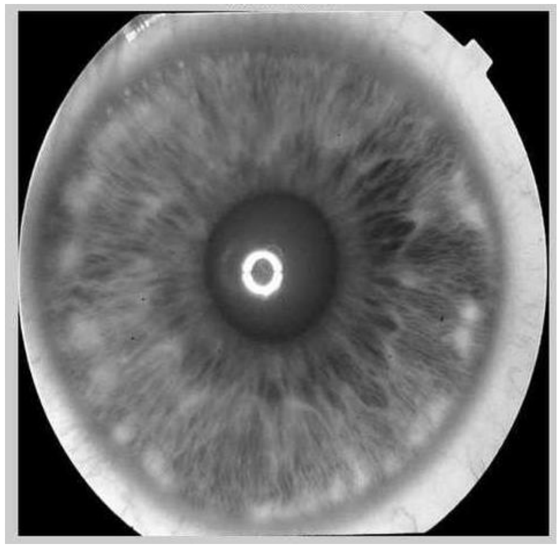


Figure 26 Image Histogram Equalization

The above diagram represents the third step which is histogram equalization.

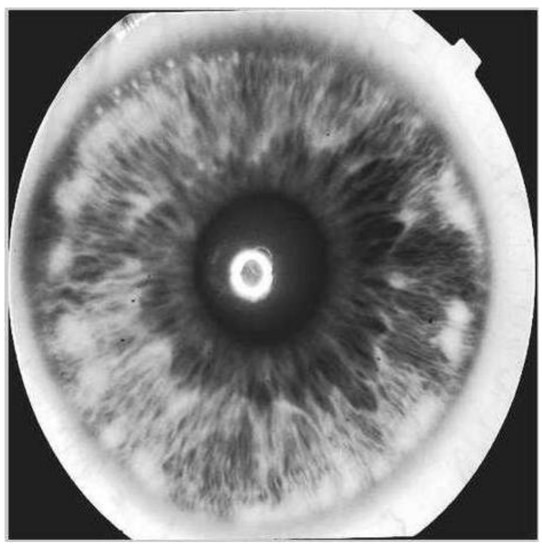


Figure 27 Gaussian Filtering

After Image equalization we apply gaussian filtering, the above image shows the results of gaussian filtering.

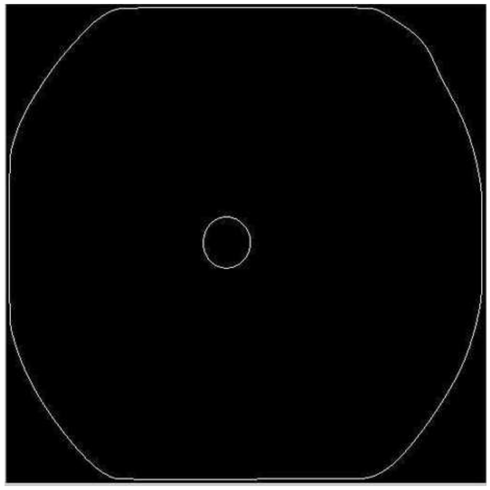


Figure 28 Canny Edge Detection

After gaussian filtering, the last step is applying canny edge detection. The above results show the basic results of iris recognition.

We have tested the iris recognition algorithm using 5 steps

The first step is segmentation using Hough transform, the second step is normalizing the image using Doughman rubber sheet model, the third step is image recognition using wilds method, the fourth step is encoding of significant feature using 1-D

4.2 Iris Recognition using Gabor Filtering

We have tested the results of Iris recognition using Gabor filtering, Gabor wavelet is based two components, one is complex sinusoidal and the other one is gaussian envelop.

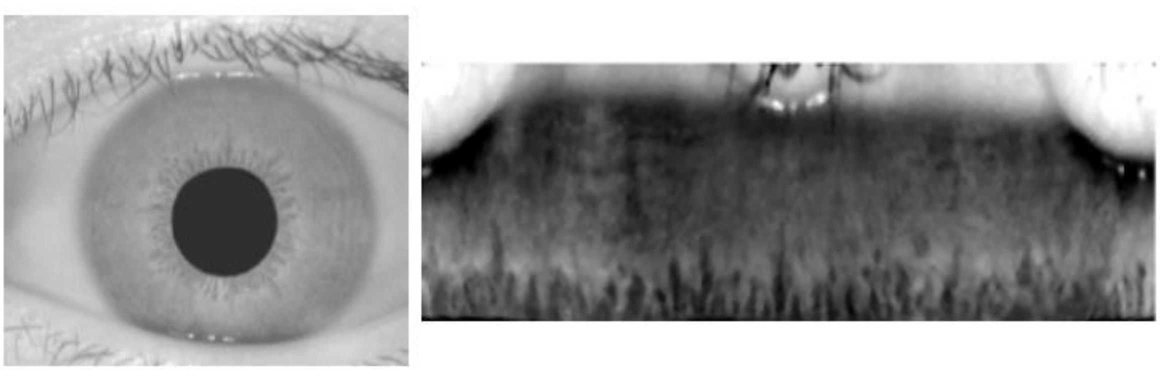


Figure 29 image with unrolled image detail

Figure shows both images, one is eye image which is shown on the left side and the other one is unrolled eye image which is shown on the right side.

Iris is basically based on two parts , one is real part and the other one imaginary part

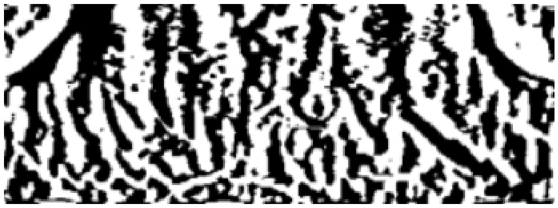


Figure 30 Real part of Image

The above figure shows the real part of iris image

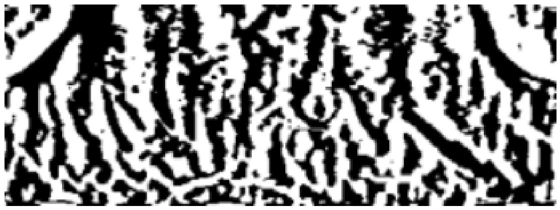


Figure 31   Imaginary part of image

This figure shows the imaginary part of iris

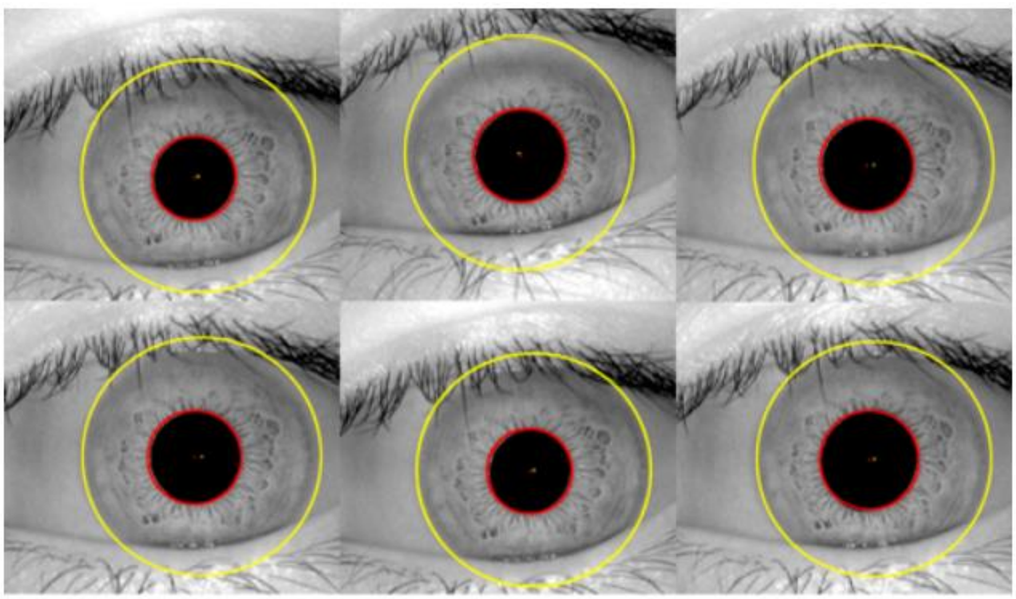


Figure 32 6 Different images of same eye

The above figure shows the 6 images of same eye, this algorithm gives 98% of results while we have tested it on 52 images. This is because the color of iris is uniform and there is good contrast of other features.

4.3 Doughman Algorithm segmentation

We have tested the Doughman algorithm for the segmentation of iris image and verify that whether it is iris image or not.

Figure 33 Taking input Image

The very first step is image acquisition in which we take the original image with good quality pixels.

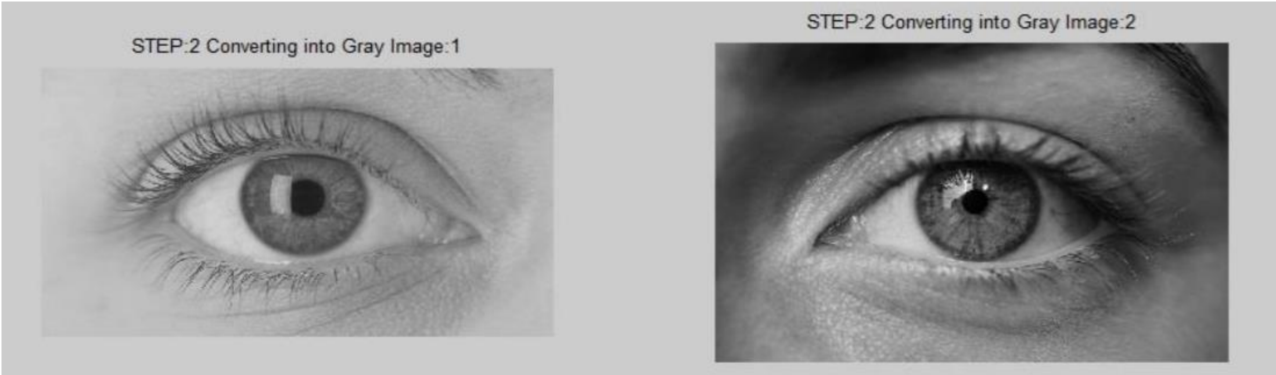


Figure 34 Gray scale conversion

We convert both the images into gray scale

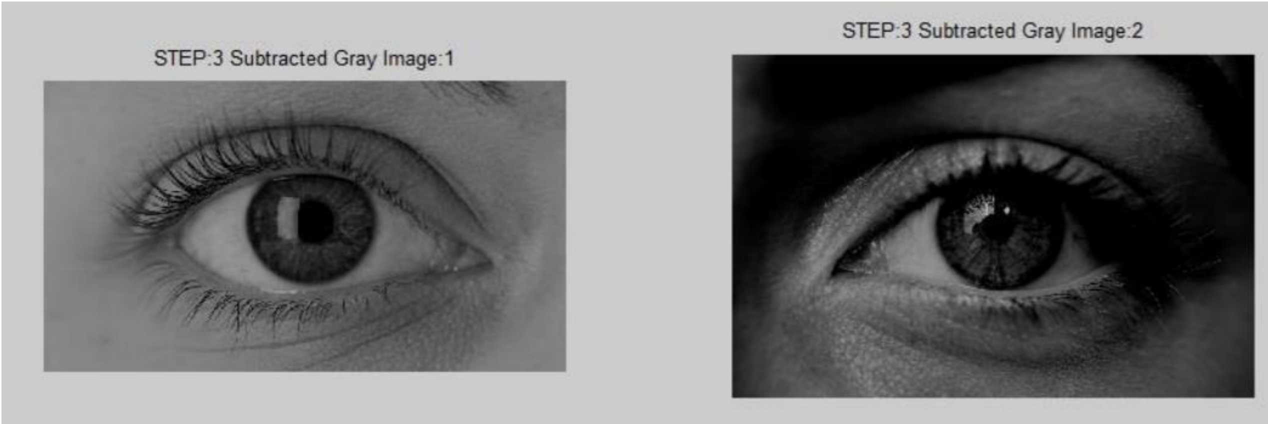


Figure 35 Subtracted gray scale Image

The above figure shows the subtracted gray images

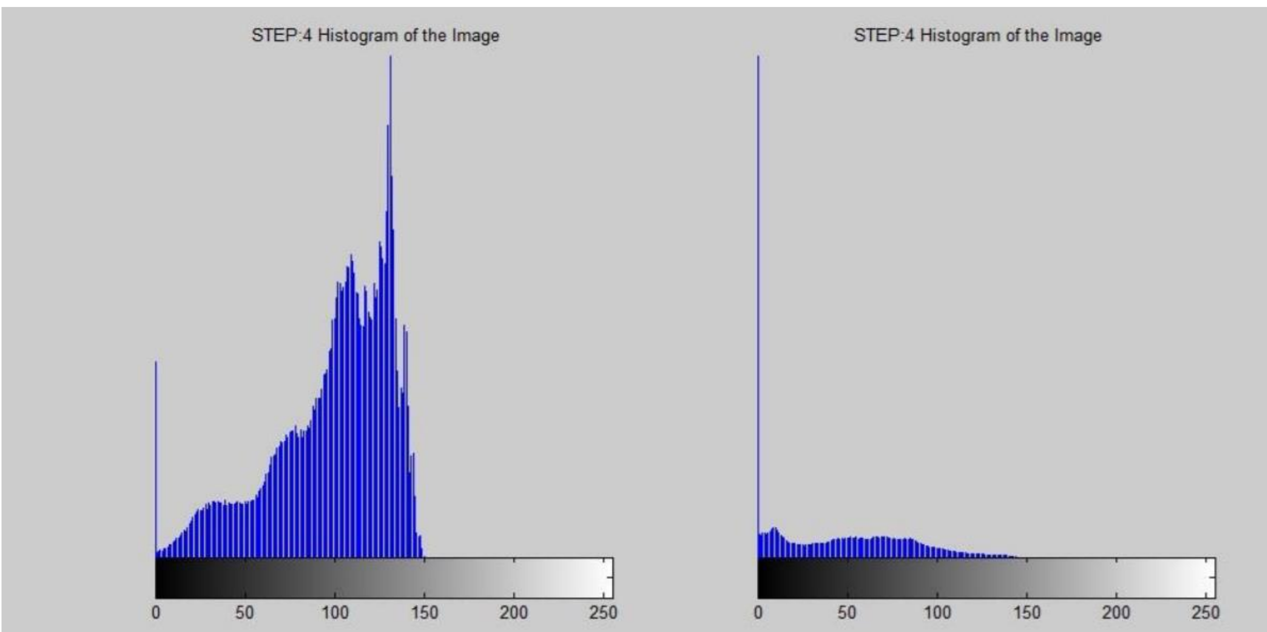


Figure 36 Histogram of both images

This figure shows the results of histogram

Figure 37 Cropped Images

In step 5 we crop the image

Figure 38 Resize images

In step 6 both the images have been resized.

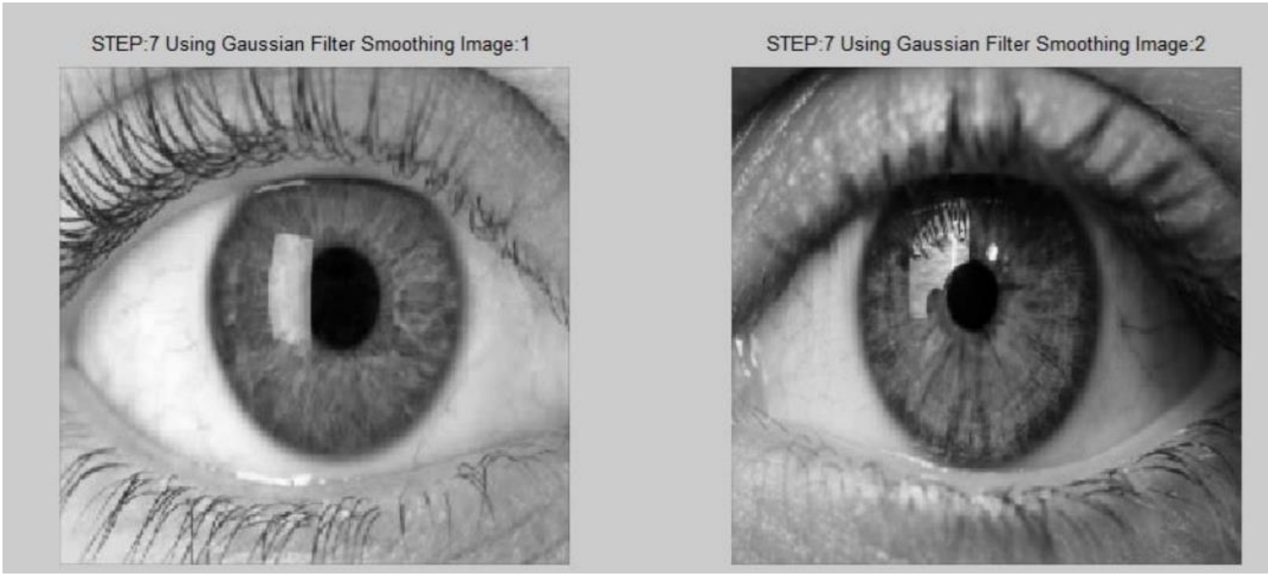


Figure 39 Gaussian filter

In step 7 we apply the gaussian filter for smoothing

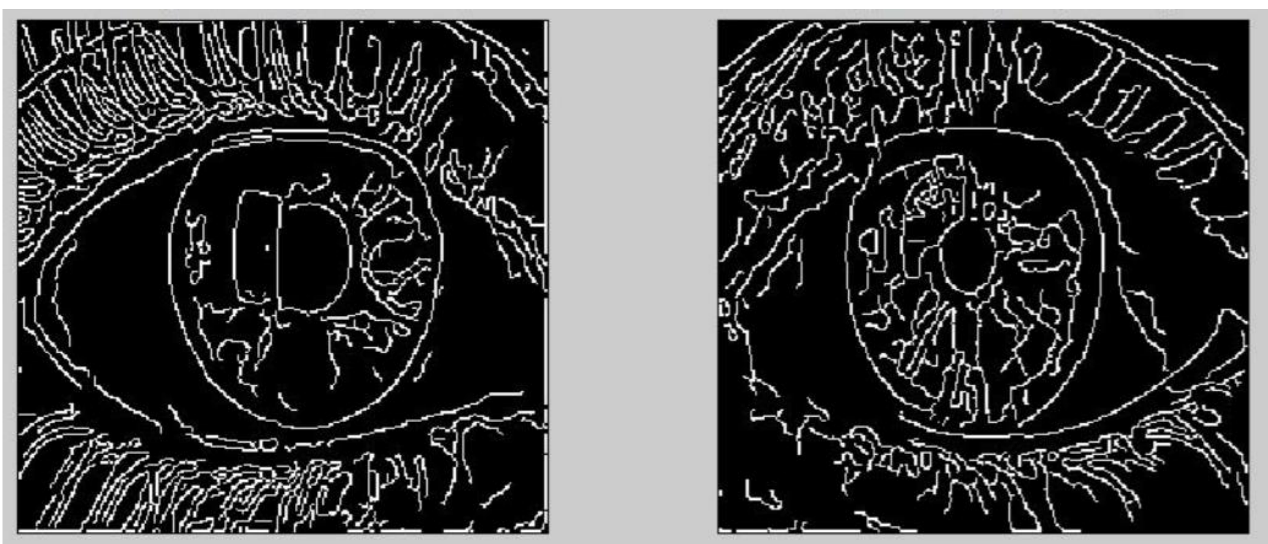


Figure 40 Canny edge detection results

In step 8 canny edge detection filter is applied

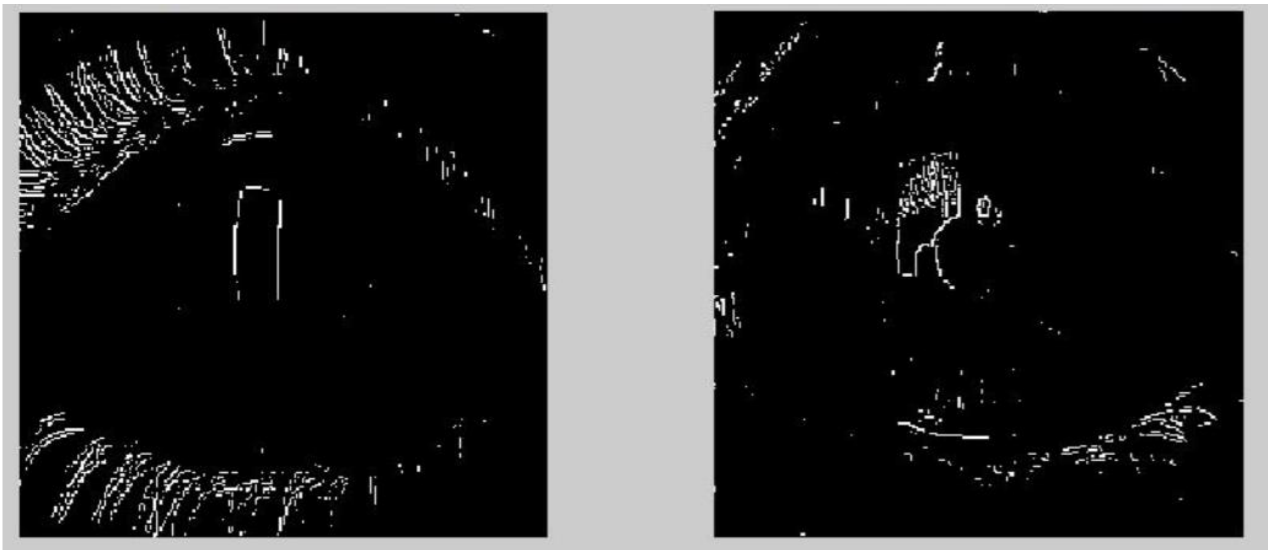


Figure 41 Sober filter results

In step 9 sober edge detection filter is applied and extract the results as shown above.

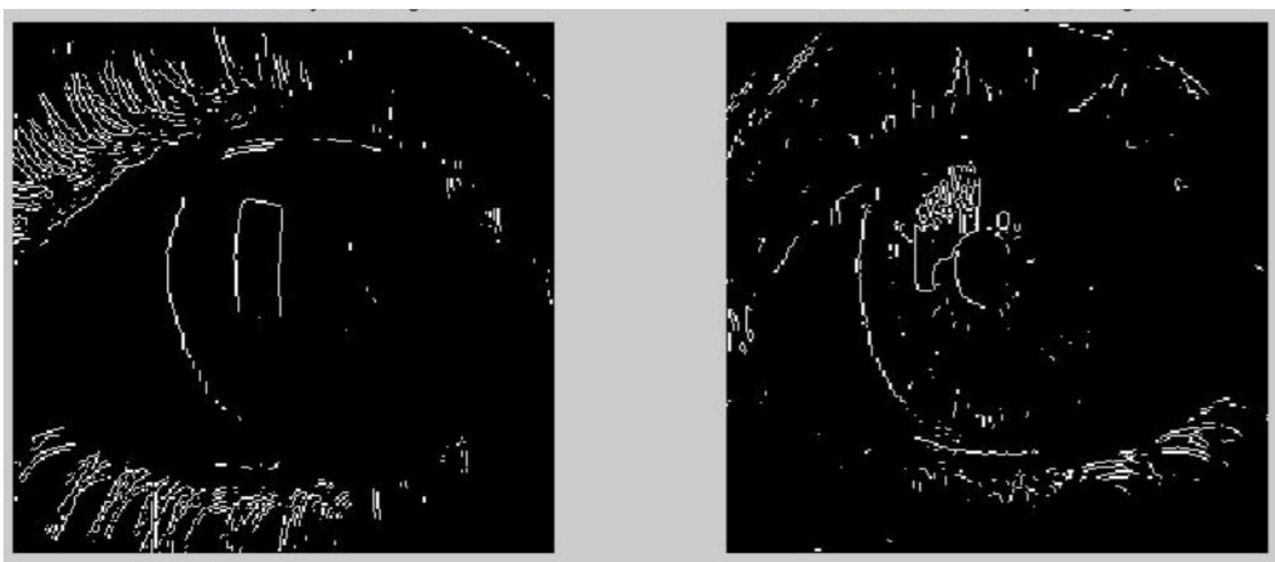


Figure 42 Gamma Adjustment

After applying filter, we apply gamma adjustment

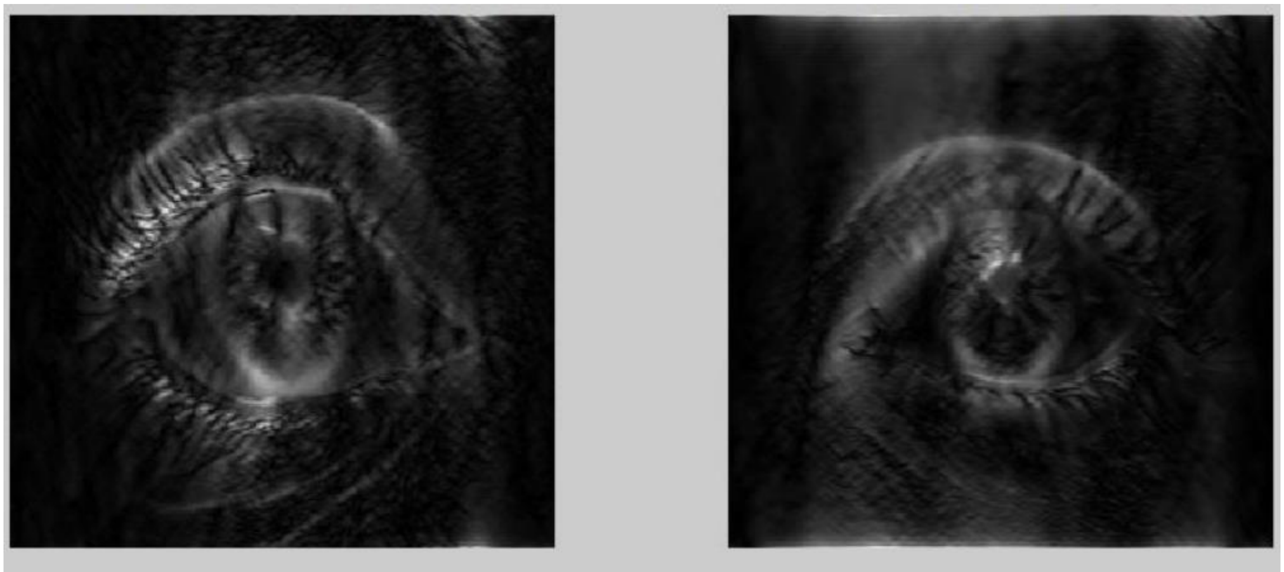


Figure 43 hysteresis Thresholding

After applying gamma filter hysteresis thresh holding is applied

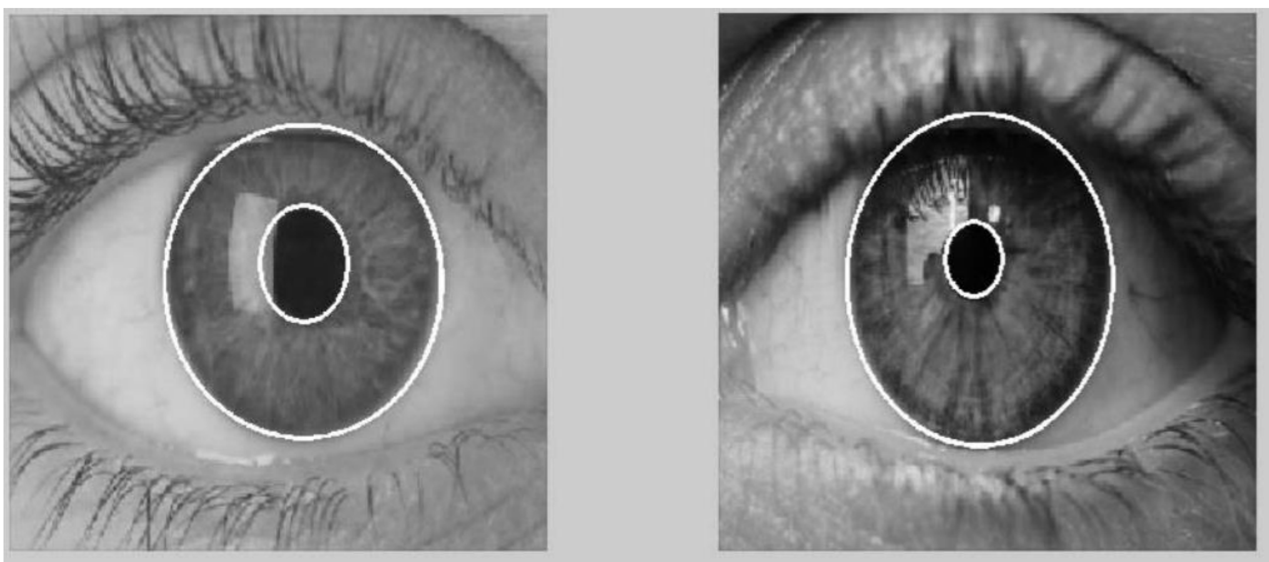


Figure 44 Hough Transform

Now we apply the Hough transform and extract the results

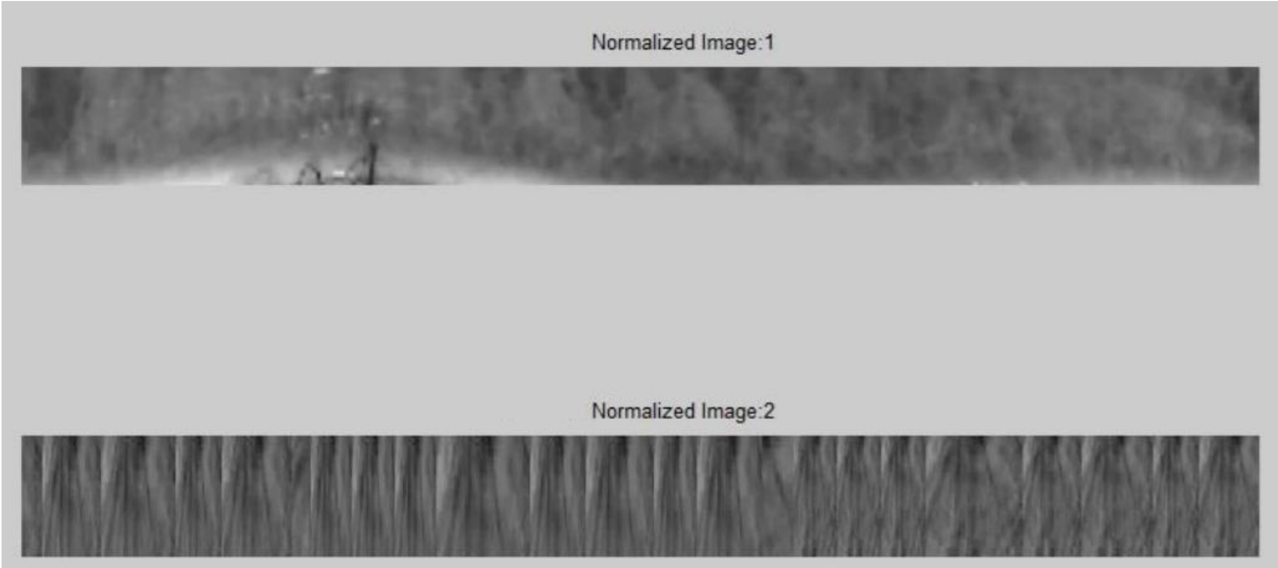


Figure 45 Normalized image results

The above figure shows the normalized images results

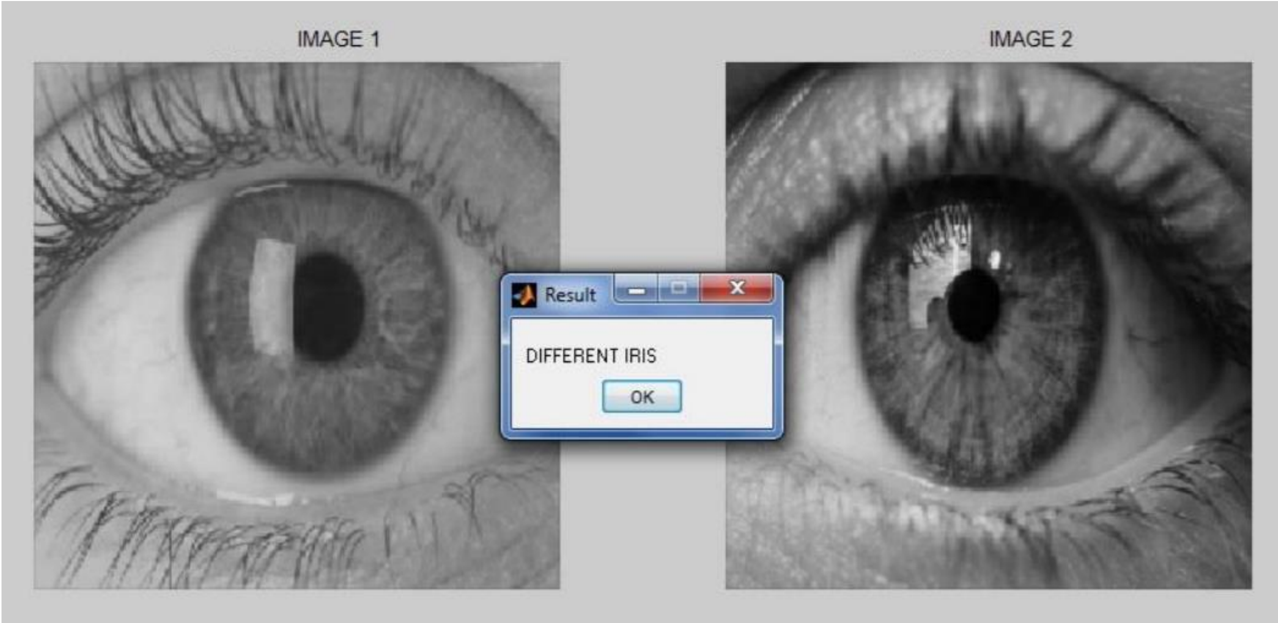


Figure 46 Final Image Results

The final output of both images is shown above which shows that the input image does not match the tested image.

Conclusions

We have performed the iris recognition-based attendance system in which we have seen that the major process iris recognition has many requirements, like we have faced a problem of high-quality camera, which is the core requirement of this project. We have applied many algorithms first of all we have tested the results of Haar cascade which is very basic and useful algorithm, by using this algorithm we just do the image processing of eyes, detect eye ball and eye retina. We first create the camera test program, then take images of eyes and after taking images train the complete system, after that we test the project. The last step of project is automatic email sending.

References

[1] S. Schuckers, P. Meyer Lopez, P. Johnson, N. Sazonova, F. Hua, R. Lazarick, C. Miles, E. Tabassi, E. Sazonov, A. Ross, and L. Hornak, "Quality--Face / Iris Research Ensemble (Q-FIRE) Data Collection Steps," 2010.
[2] S. Schuckers, P. Meyer Lopez, P. Johnson, N. Sazonova, F. Hua, R. Lazarick, C. Miles, E. Tabassi, E. Sazonov, A. Ross, and L. Hornak, "Quality--Face / Iris Research Ensemble (Q-FIRE) Dataset Overview," 2010.
[3] N. Othman, and B. Dorizzi: Impact of Quality-Based Fusion Techniques for Video-Based Iris Recognition at a Distance, in Information Forensics and Security, IEEE Transactions on , vol.10, no.8, pp.1590-1602, Aug. 2015.
[4] X. Tan, F. Song, Z.-H. Zhou, and S. Chen, "Enhanced pictorial structures for precise eye localization under incontrolled conditions," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 1621–1628.
[5] http://www.csse.uwa.edu.au/~pk/studentprojects/libor/in dex.html
[6] http://www.biometricscatalog.org/NSTCSubcommittee/
Documents/Iris%20Recognition.pdf
[7] John.Daugman@CL.cam.ac.uk [4]. J. G. Daugman: High confidence visual recognition of persons by a test of statistical independence. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 15 (1993) 1148–1161
[8]..W.W. Boles, B. Boashah: A Human Identification Technique Using Images of the Iris and Wavelet Transform. IEEE Transaction on Signal Processing Vol. 46 (1998) 1185-1188
[9]. T. Chuan Chen, K. Liang Chung: An Efficient Randomized Algorithm for Detecting Circles. Computer Vision and Image Understanding Vol. 83 (2001) 172-191
[10]. T. Chuan Chen, K. Liang Chung: An Efficient Randomized Algorithm for Detecting Circles. Computer Vision and Image Understanding Vol. 83 (2001) 172-191 [8]. Peter Kovesi, Matlab functions for Computer Vision and Image Processing. What are Log-Gabor filters ?
[11] Rubea Othman Rubea, Xue Wen Ding, ATM security system with IRIS recognation and GSM module volume 7, tianjin china, 2018
[12] P.S Bhagat, Prof D.S Shilwant, Prof S.P Kharde, Prof Amol A.Shirsath, IRIS based attendance system volume 4, Mahrashtar India, 2015
[13] Seifedine kadry & Mohammad Smaili, Wireless attendance mgmt system based on IRIS recogination, volume 5, Lebnon, 2018
[14] Oluwagbemiga Shoewu, Olusegun Omitola,Iris Recognition Technology: Implementation, Application, and Security
Consideration,2013
[15] k. Mahesh, Hari Hara Brahmal, Dr. G. Kodanda Ramaiah, ATM based recognition technique on iris Technology with GSM module, International Journal of Scientific Engineering and Technology Research, ISSN 2319-8885, Vol.04, Issue.51, December-2015
[16]https://www.researchgate.net/publication/264136154_Iris_recognition_Threat_analysis_at_the_user_interface_level
[17] https://bomamarketing.com/2017/01/19/how-do-i-build-an-effective-email-database-for-my-business/
[18] https://electrosome.com/interfacing-gsm-module-arduino/