Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

FALL SEMESTER 2021-2022

IRIS RECOGNITION SYSTEM TO DETECT BOGUS FARMERS

REVIEW – 2:

COURSE CODE : SWE1015

SLOT : G2

COURSE : BIOMETRIC SYSTEMS

FACULTY : PROF. RAMYA G

TEAM MEMBERS:

SATHISH KUMAR P - 18MIS0383

ESWAR G - 19MIS0018

ASHWIN PAI - 19MIS0019

JEEVADHARSHINI S - 19MIS0063

VARUN P - 19MIS0128

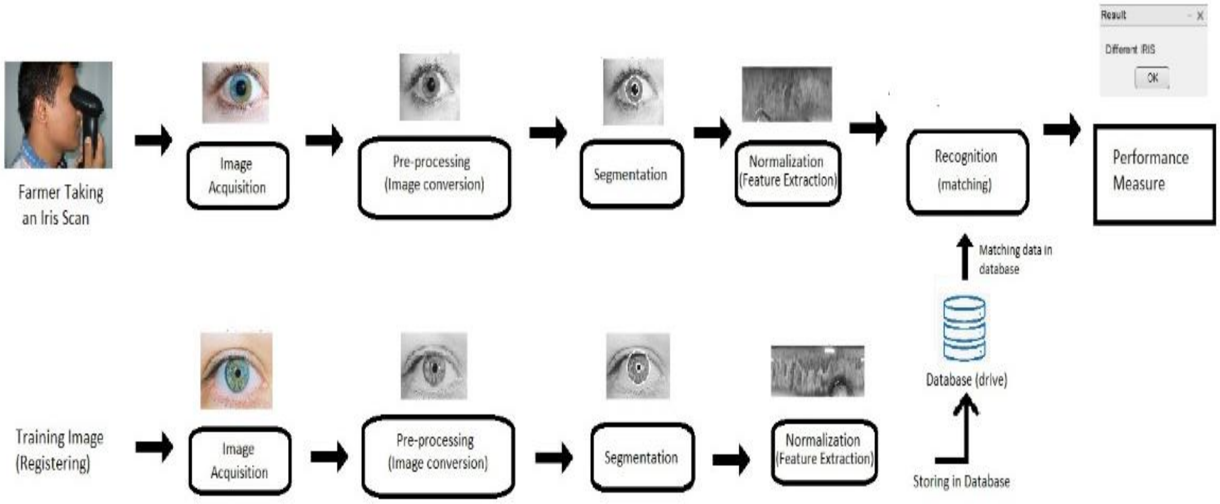
KATKAM SHANMUKH AKUL - 19MIS0130

HEMALATHA M - 19MIS0219

SHARMILA B - 19MIS0236

AKHILA K - 19MIS0350

SYSTEM ARCHITECTURE:



SYSTEM REQUIREMENTS:

MATLAB Online is the online open source tool used to perform this implementation.

Additional libraries(Add-Ons) required to perform the implementation are given below:

- Image Processing ToolBox
- Houghs Transform For Circles
- Pupil/Limbus Detection and Daugmann Normalisation
- Hysteresis Thresholding for 2D images
- Gabor Feature Extraction

Sample iris focussed images from a random Database is used and stored in MATLAB DRIVE as a storage component for local purpose.

DESCRIPTION:

Step:0 image acquisition process

In this step, Image captured and image for comparison are selected from the database (MATLAB DRIVE)

Step:1 (PRE-PROCESSING)

Output of the raw image is shown for initial identification

Step:2

Raw image is converted into Grayscale image for minimising the computational requirements, time and t increase accuracy of the image components. Also, grayscale conversion is the basic necessary for image pre processing and convert image for pixel friendly.

Step:3

In case, if the image has to much brightness and more contrast after converting it into grayscale image, it is subtracted to much lower brightness for convenience.

Step:4

Initial histogram of the two images are generated and compared which represents the initial pixel statistics for comparison. Image is first converted to a 8-bit for the pixels representation.

Step:5 (SEGMENTATION and LOCALIZATION)

Images are cropped to eliminate unwanted details shown in the image. Cropping the image requires to note down the co-ordinates from both ends which covers the iris and pupil area.

Step:6

Images are resized since cropping changes the whole resolution of the image. Images are resized to 256x256 resolution so that they can be represented as unsigned 8-bit image.

Step:7

The re-sized images are smoothened using a Gaussian filter since smoothening helps in better composition and clarity of the image and removes unwanted detail.

Step:8

Caany Filter Edge Detection remains the best edge detection method out of all others. Images are being used with canny edge detection to detect the edges of the iris and pupil perimeter. Intensities can be raised according to our requirement. Also, edge detection in this automatically uses hysteresis thresholding to get the most out of the details and bring out the clearer edge.

Step:9

Gamma Correction used to correct the canny edge detected image to adjust the contrast and clarity of the image. It is optional and is upto the user’s requirement for effective segmentation

Step:10

Using Sobel filter which is another edge detection method to identify the edges. It is more intense than canny edge detection and could find more edges present in the eye image.

Now that the edges are well detected, hence, the images are suitable and eligible for segmenation process. Segmentation involves extracting the area between the borders of iris and pupil of the eye image.

Step:11

Hough’s Transform is used to detect the circular edges present as the shape of iris and pupil. Noting down the co ordinates of the center position of the iris and pupil is necessary to find out the radius of them required for Hough’s Transform to detect the circles. Hough’s Transform remains an excellent method for segmenting the iris and pupil region separately.

Requirement: co ordinate of the centers, diameter/radius of iris and pupil to draw an arc throughout and draw the lines of circle covering the iris and pupil perimeter.

The main advantage of the Hough transform technique is that it is tolerant to gaps in feature boundary descriptions and is relatively unaffected by image noise and localise irises very effectively.

The Hough transform method requires the threshold values to be chosen for edge detection, and this may result in critical edge points being removed, thus resulting in failures to detect circles/arcs.

Step:12

Normalisation of the images are carried out after Hough’s Transform. Normalization produces a 2D array with horizontal dimensions of angular resolution and vertical dimensions of radial resolution. The area covered by the concentric circle drawn using hough’s transform depicts the iris region which is required excluding the pupil region. The iris region of both the images are extracted from the whole eye image and normalised so they are represented flatly. Daugman’s Rubber sheet normalisation method is used here for normalising the images. The images are mapped to same size of rectangular region. “Rubber Sheet” is produced linearly stretching and compressing the iris region to a standard size frame for both images for feature extraction and matching.

Required Input Parameters: Co ordinates of the centers of pupil and iris, Radii of Iris, Radii of pupil

Images are required to be saved in the MATLAB DRIVE for selecting them for comparison and Matching in the next step as for Hamming Distance Calculation.

Step:13

Normalised images are selected from the MATLAB DRIVE which had been saved in the drive in the previous step. Enhancement of the Normalised Iris images takes place here using Histogram Equalisation for better quality and enhancement of the image. Enhancement is also needed for Feature extraction and encoding of the normalised iris image for comparison.

Step:14

The enhanced normalised images are allowed for feature extraction using Gabor wavelet Filter which takes the wavelength as pixels. So, giving the pixels as 256x256 which will create the magnitude and phase of the texture filter with required orientation. Default orientation is given as 0◦. Now the features are extracted using Gabor Filter, the iris images need to get encoded to binary number 8 bits for measuring the hamming distance.

Step:15

The feature extracted enhanced normalised images are encoded using a MATLAB function(dec2bin) which converts decimal matrix to Binary numbers with required 8 bits. Now, the images are converted to binary 8 bit numbers, they are ready for comparison and matching using Hamming distance.

HAMMING DISTANCE:

The Hamming Distance was chosen as a matching metric, which gave a measure of how many bits disagreed between the two images. When the hamming distance of two images is calculated, one template is shifted left and right bitwise and a number of hamming distance values are calculated from successive shifts, in order to account for rotational inconsistencies. False Acceptance rate is calculated through this process. Generally, Daugman's Threshold goes with 0.5. Here, it is minimised to 0.3 after experimenting with different images.

Algorithm for hamming distance is given in the code below.

0 <= HD <= 0.3 = SAME IRIS

0.3 < HD <= 0.4 = UNCERTAIN, use different iris images or please run the whole test again

HD > 0.4 = DIFFERENT IRIS

IMPLEMENTATION:

The tool we have used for implementing iris recognition system is MATLAB and below represents the 100% implementation of our project.

Code:

```
% STEP: 0 Image Aquisition Process:-
```

```matlab
clc; close all; warning off;

[FileName1,FilePath1] = uigetfile('*.jpg','Select captured Iris Image');
file1= fullfile(FilePath1, FileName1);

[FileName2,FilePath2] = uigetfile('*.jpg','Select the Iris Image for comparison');
file2= fullfile(FilePath2, FileName2);


% STEP:1 IMAGE AQUISITION

i=imread(file1);
subplot(1,2,1);
imshow(i);
title('STEP:1 Image Aquisition(Image:1)');

ii=imread(file2);
subplot(1,2,2);
imshow(ii);
title('STEP:1 Image Aquisition(Image:2)');

figure();



% STEP:2 GRAY SCALE CONVERSION

g =rgb2gray(i);
subplot(1,2,1);
imshow(g);
title('STEP:2 Conversion Gray Image:1');

gg=rgb2gray(ii);
subplot(1,2,2);
imshow(gg);
title('STEP:2 Conversion Gray Image:2');

figure();

% STEP:3 Subtraction of image

k = imread(file1);
v= rgb2gray(k);
v = imsubtract(v,60);
subplot(1,2,1);
imshow(v);
title('STEP:3 Subtracted Gray Image:1');

kk = imread(file2);
vv =rgb2gray(kk);
vv = imsubtract(vv,60);
subplot(1,2,2);
imshow(vv);
title('STEP:3 Subtracted Gray Image:2');

figure();

% STEP:4 HISTOGRAM of the IMAGE
% Convert the image to unsigned 8 bit image and plot the histogram
```

```
z=double(g);
subplot(2,1,1);
imhist(g);
title('STEP:4 Histogram Image:1');

zz=double(gg);
subplot(2,1,2);
imhist(gg);
title('STEP:4 Histogram Image:2');

figure();


% STEP:5 CROPPED IMAGE

c=imcrop(g,[138 96 787 402]);
subplot(1,2,1);
imshow(c);
title('STEP:5 Cropped Image:1');

cc=imcrop(gg,[78 128 893 573]);
subplot(1,2,2);
imshow(cc);
title('STEP:5 Cropped Image:2');

figure();


% STEP:6 RESIZED IMAGE

r=imresize(c,[256,256],'nearest');
subplot(1,2,1);
imshow(r);
title('STEP:6 Resized Image:1');

rr=imresize(cc,[256,256],'nearest');
subplot(1,2,2);
imshow(rr);
title('STEP:6 Resized Image:2');

figure();

% STEP:7 IMAGE SMOOTHING

s= fspecial('gaussian',3);
f = imfilter(r,s);
subplot(1,2,1);
imshow(f,[]),title('STEP:7 Gaussian Filter Smoothing Image:1 ');

ss= fspecial('gaussian',3);
ff = imfilter(rr,ss);
subplot(1,2,2);
imshow(ff,[]),title('STEP:7 Gaussian Filter Smoothing Image:2');

figure();

%%%Image Segmentation Process:-
```

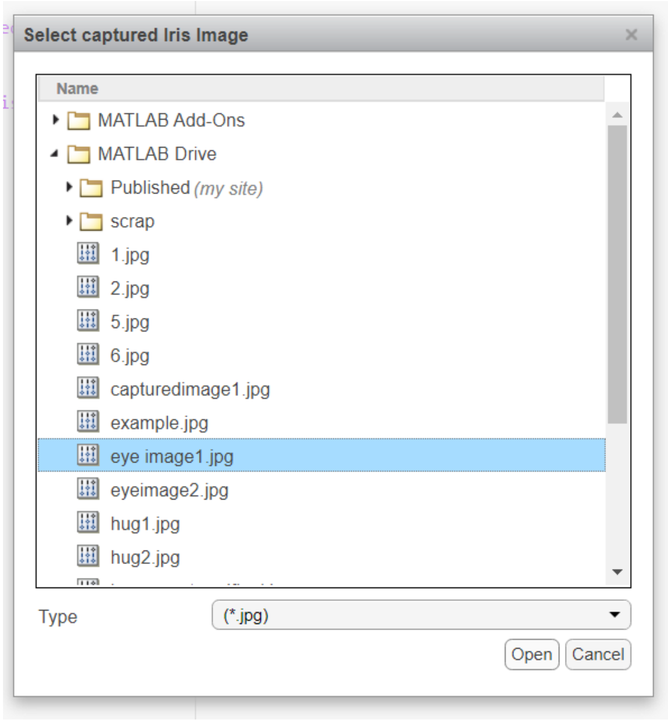
```
% STEP:8 CANNY EDGE DETECTION

e=edge(f,'canny');
subplot(1,2,1);
imshow(e);
title('STEP:8 Canny Filter Edge Detection Image:1');

ee=edge(ff,'canny');
subplot(1,2,2);
imshow(ee);
title('STEP:8 Canny Filter Edge Detection Image:2');

figure();



% STEP:9 GAMMA CORRECTION
%Adjust the Gamma to 0.8

S=edge(f,'sobel');
u=double(S);
subplot(1,2,1);
y= imadjust(u,[],[],0.02);
imshow(y);
title('STEP:9 Gamma Adjusted Image:1');

SS=edge(ff,'sobel');
uu=double(SS);
subplot(1,2,2);
yy= imadjust(uu,[],[],0.02);
imshow(yy);
title('STEP:9 Gamma Adjusted Image:2');
figure();

%STEP:10  EDGE DETECTION using Sobel filter

S1=edge(f,'sobel',0.02);
subplot(1,2,1);
imshow(S1);
title('STEP:10 Edge Detection by Sobel Filter Image:1');

SS1=edge(ff,'sobel',0.02);
subplot(1,2,2);
imshow(SS1);
title('STEP:10 Edge Detection by Sobel Filter Image:2');

figure();


%STEP:11 HOUGH'S TRANSFORM TO DETECT CIRCLES     %if layer is a rgb image turn
into grayscale

i=imread(file1);
g=rgb2gray(i);
subplot(1,2,1);
imshow(g);
[centersiris1, radiiiris1] = imfindcircles(g,[74 75],'ObjectPolarity','dark',
'Sensitivity',0.99,'EdgeThreshold',0.1,'Method','TwoStage');
```

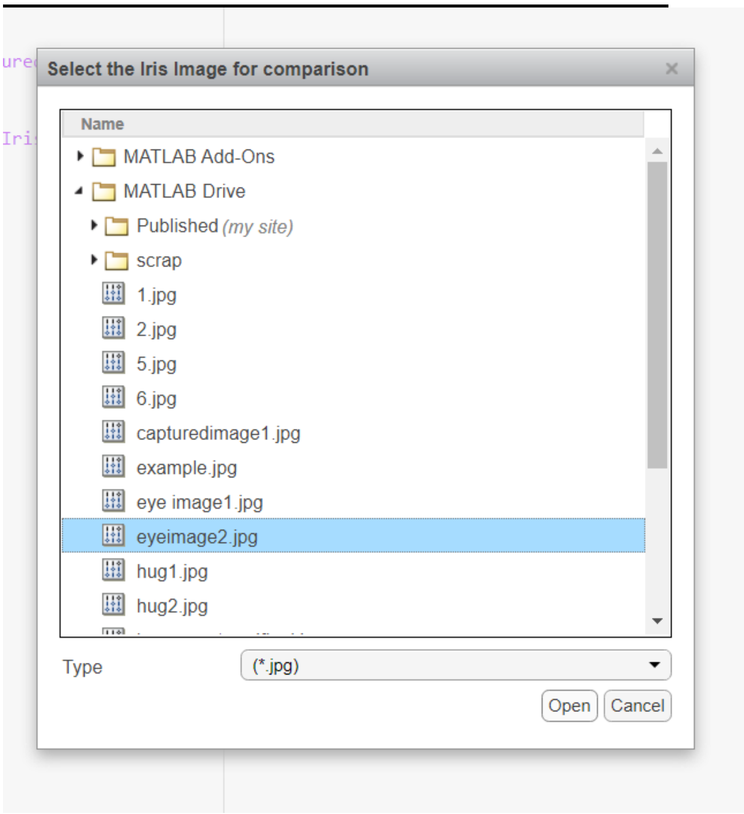
```
[centerspupil1, radiipupil1] = imfindcircles(g,[35 36],'ObjectPolarity','dark',
'Sensitivity',0.96,'EdgeThreshold',0.1,'Method','TwoStage');
viscircles(centersiris1,radiiiris1,'color','w');
viscircles(centerspupil1, radiipupil1,'Color','w');
title("STEP:11 Houghs Transform (image1)");


i=imread(file2);
gg=rgb2gray(i);
subplot(1,2,2);
imshow(gg);
[centersiris2, radiiiris2] = imfindcircles(gg,[130 135],'ObjectPolarity','dark',
'Sensitivity',0.99,'EdgeThreshold',0.1,'Method','TwoStage');
[centerspupil2, radiipupil2] = imfindcircles(gg,[27 28],'ObjectPolarity','dark',
'Sensitivity',0.96,'EdgeThreshold',0.1,'Method','TwoStage');
viscircles(centersiris2,radiiiris2,'color','w');
viscircles(centerspupil2, radiipupil2,'Color','w');
title("STEP:11 Houghs Transform (image2)");


%STEP :12 NORMALIZATION OF IMAGES
% Load the image
img1 = imread(file1);
% Input parameters
xPosPupil= 264;
yPosPupil= 168;
rPupil= 35;
xPosIris= 264;
yPosIris= 168;
rIris =74;
% Normalize the iris region according to daugmans model
ir1 = rubberSheetNormalisation( img1, xPosPupil, yPosPupil, rPupil , xPosIris ,
yPosIris , rIris,'DebugMode', 1,'UseInterpolation', 0);
figure();
% Show Resulting image
subplot(2,1,1);
imshow(ir1);
title("STEP:12 Normalised Image:1");
imsave();


img2 = imread(file2);
% Input parameters
xPosPupil= 390;
yPosPupil= 270;
rPupil= 27;
xPosIris= 390;
yPosIris= 270;
rIris =130;
% Normalize the iris region according to daugmans model
ir2 = rubberSheetNormalisation( img2, xPosPupil, yPosPupil, rPupil , xPosIris ,
yPosIris , rIris,'DebugMode', 1,'UseInterpolation', 0);
% Show Resulting image
subplot(2,1,2);
imshow(ir2);
title("STEP:12 Normalised Image:2");
imsave();

%STEP:13,14,15: ENHANCEMENT, EXTRACTION AND MATCHING USING HAMMING DISTANCE
```

```
[FileName1,FilePath1] = uigetfile('*.png','Select normalised image no.1');
file1= fullfile(FilePath1, FileName1);

[FileName2,FilePath2] = uigetfile('*.png','Select the normalised Image for 2
,comparison');
file2= fullfile(FilePath2, FileName2);

%%Enhancement of normalised iris images%%
image1 = imread(file1);
image2 = imread(file2);
J= histeq(image1);
J2= histeq(image2);
figure();
subplot(2,1,1);
imshow(J);
title("STEP:13-Enhanced iris image 1");
subplot(2,1,2);
imshow(J2);
title("STEP:enhanced iris image 2");

%%Feature Extracrtion using wavelet pixels GaborFilter
[mag1,phase1] = imgaborfilt(J,256,0);
[mag2,phase2] = imgaborfilt(J2,256,0);

%%Hamming distance between the normalised images
x1 = dec2bin(phase1,8);
y1 = dec2bin(phase2,8);

hd=0;
for i=1:length(x1)
    if(x1(i)~=y1(i))
        hd=hd+1;
        disp(hd);
    end
end
hd = hd/100000;

if hd < 0 && hd <= 0.2
h = msgbox('Same IRIS','Result');
elseif hd >= 0.4
h = msgbox('Different IRIS','Result');
elseif hd <=0.4 && hd >= 0.3
h = msgbox('Uncertain, Please run test again or change iris images','Result');
elseif hd ==0
h = msgbox('Same IRIS','Result');
end
```

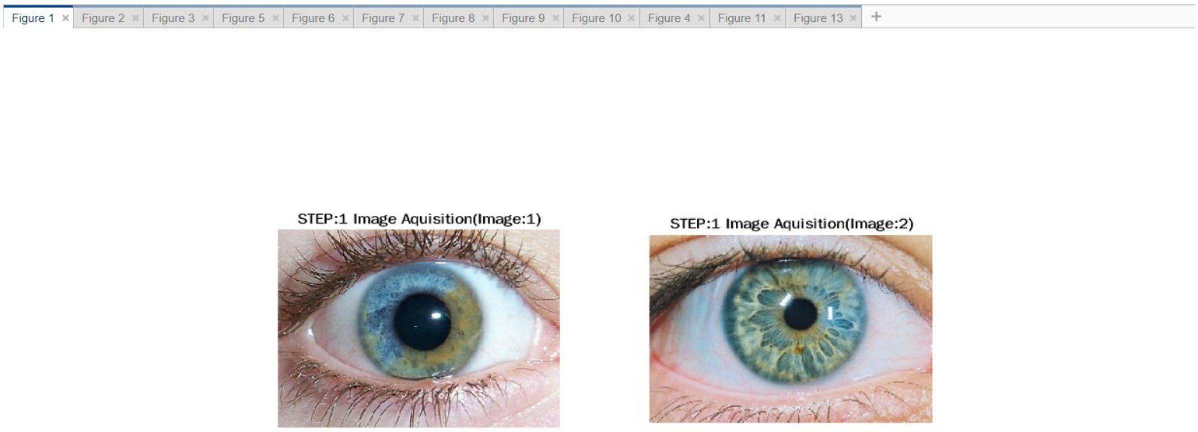
OUTPUT:

STEP:0 - SELECTING CAPTURED IMAGE AND IMAGE FOR COMPARISON

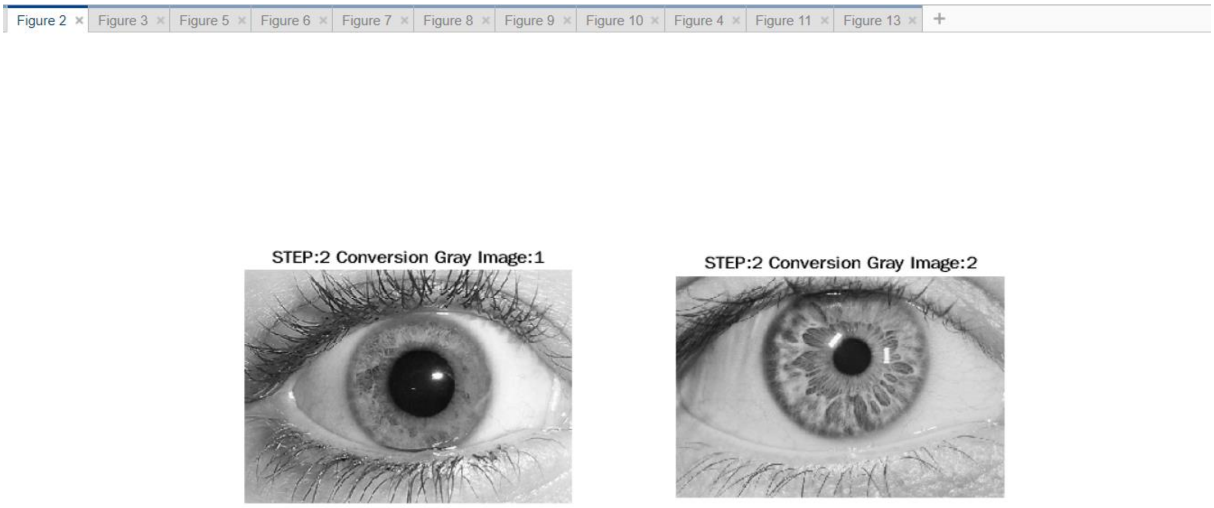




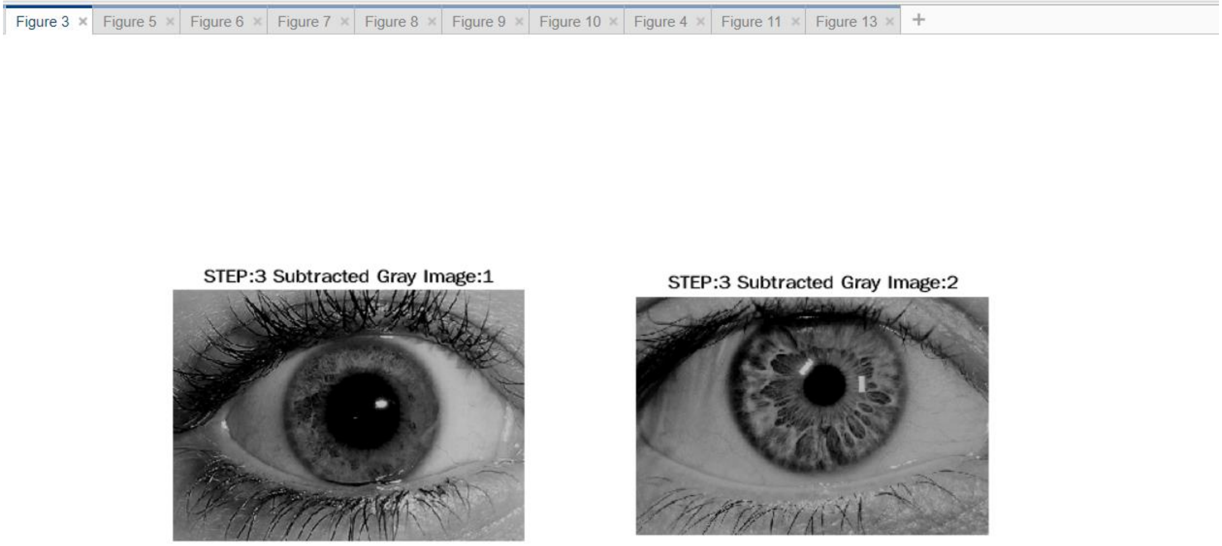
STEP:1 – IMAGE ACQUISITION PROCESS



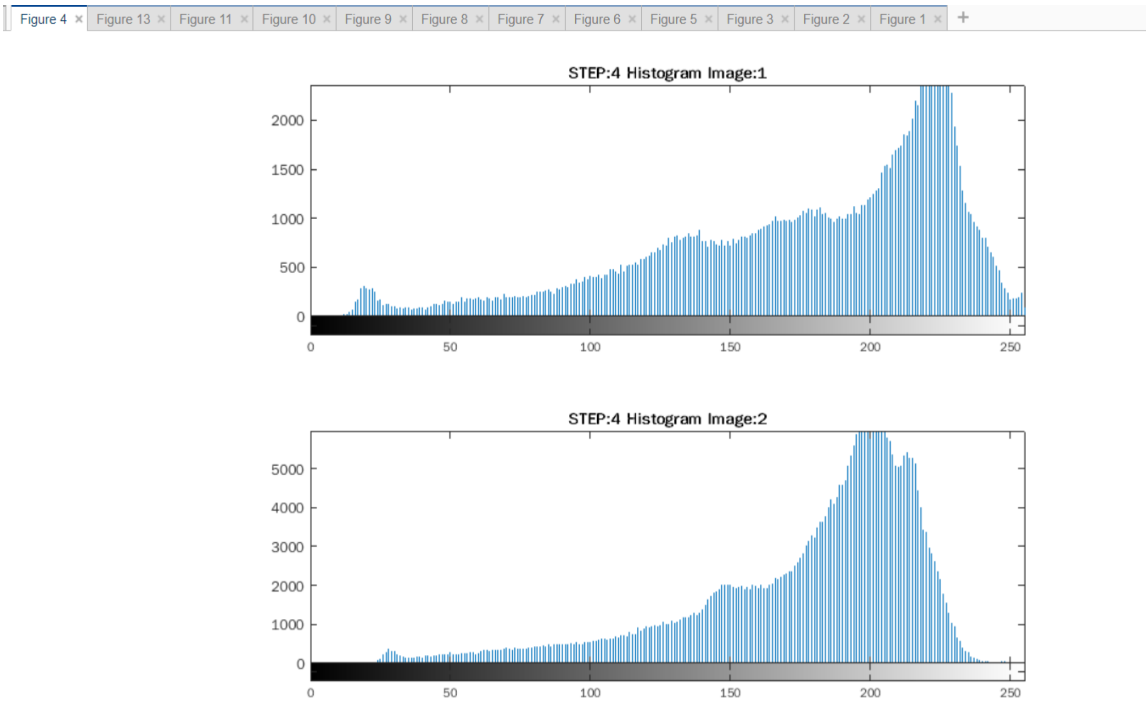
STEP:2 – GRAYSCALE CONVERSION



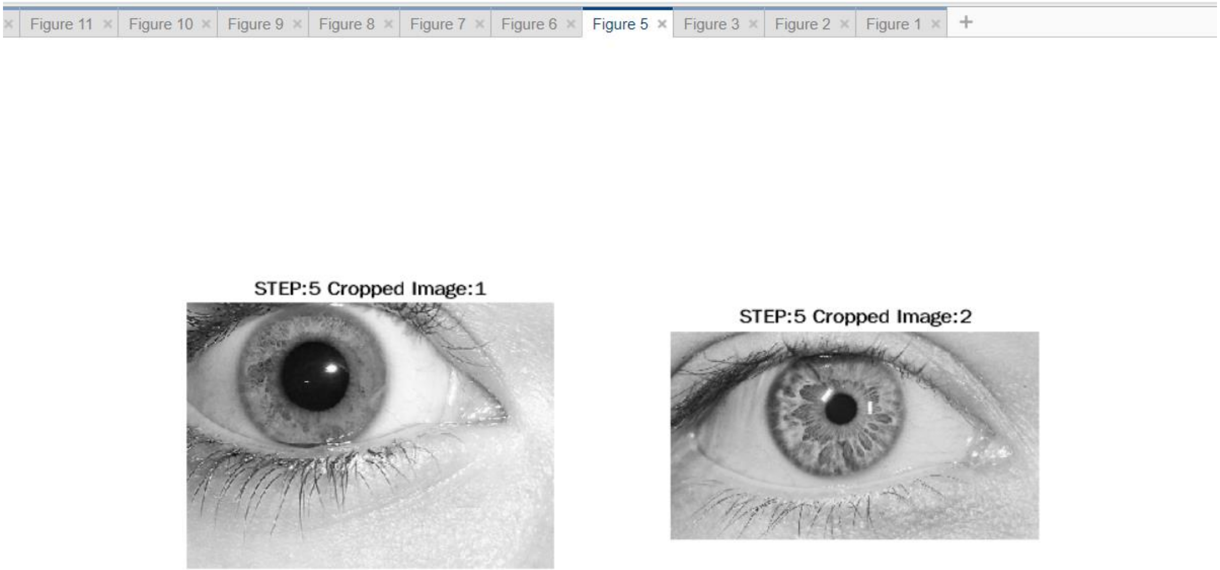
STEP:3 – SUBTRACTION OF GRAYSCALE IMAGE



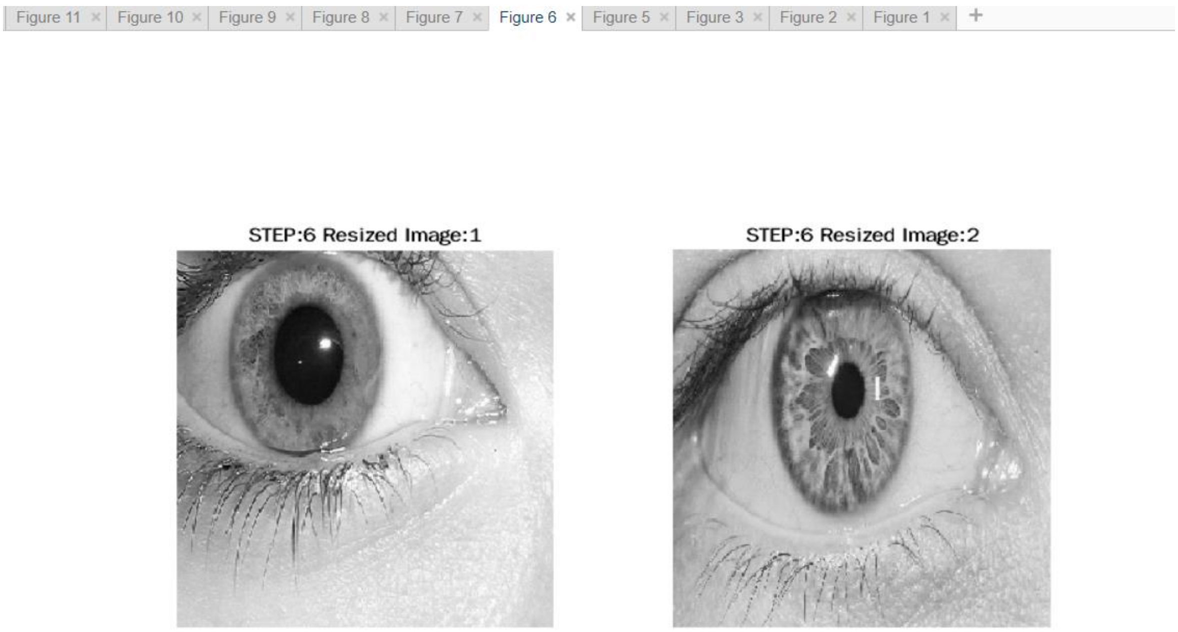
STEP:4 – HISTOGRAM OF IMAGES



STEP:5 – CROPPED IMAGE



STEP:6 – RESIZED IMAGE



STEP:7 – SMOOTHENING OF IMAGE USING GAUSSIANFILTER

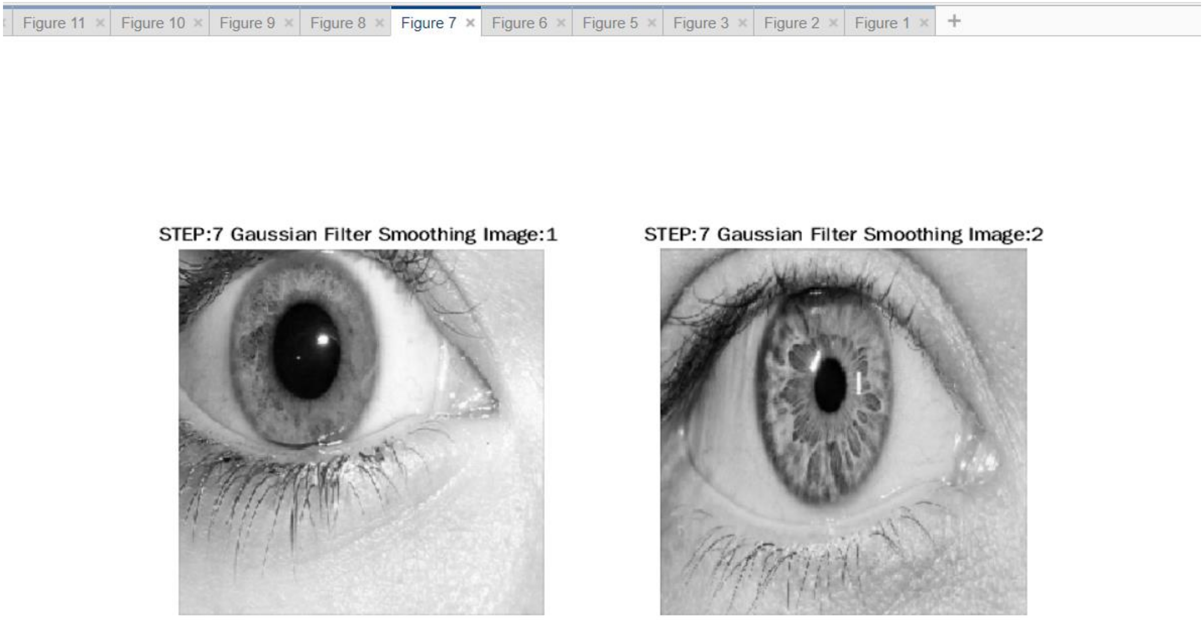
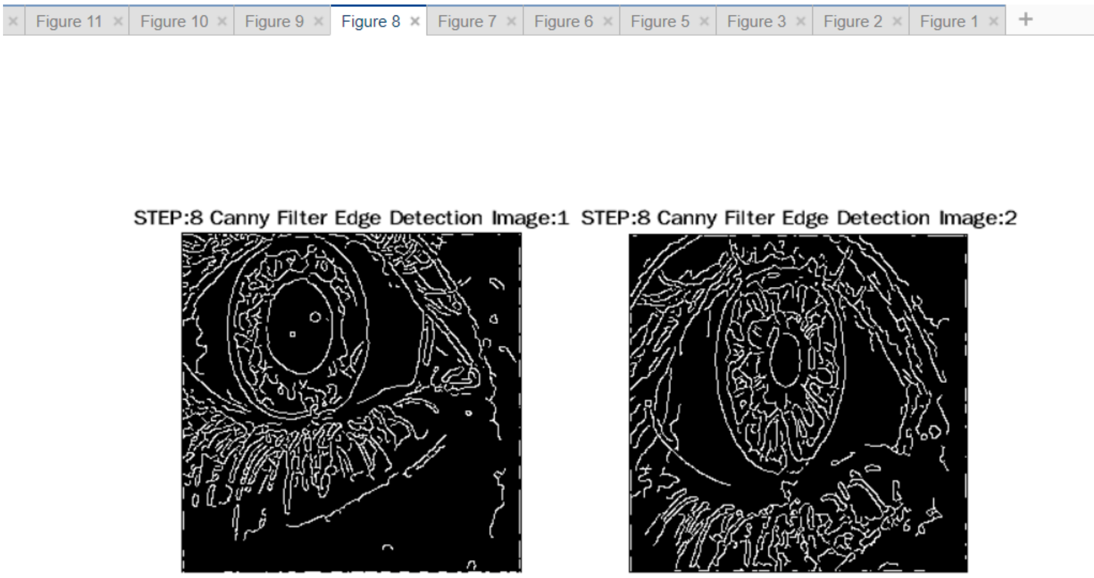
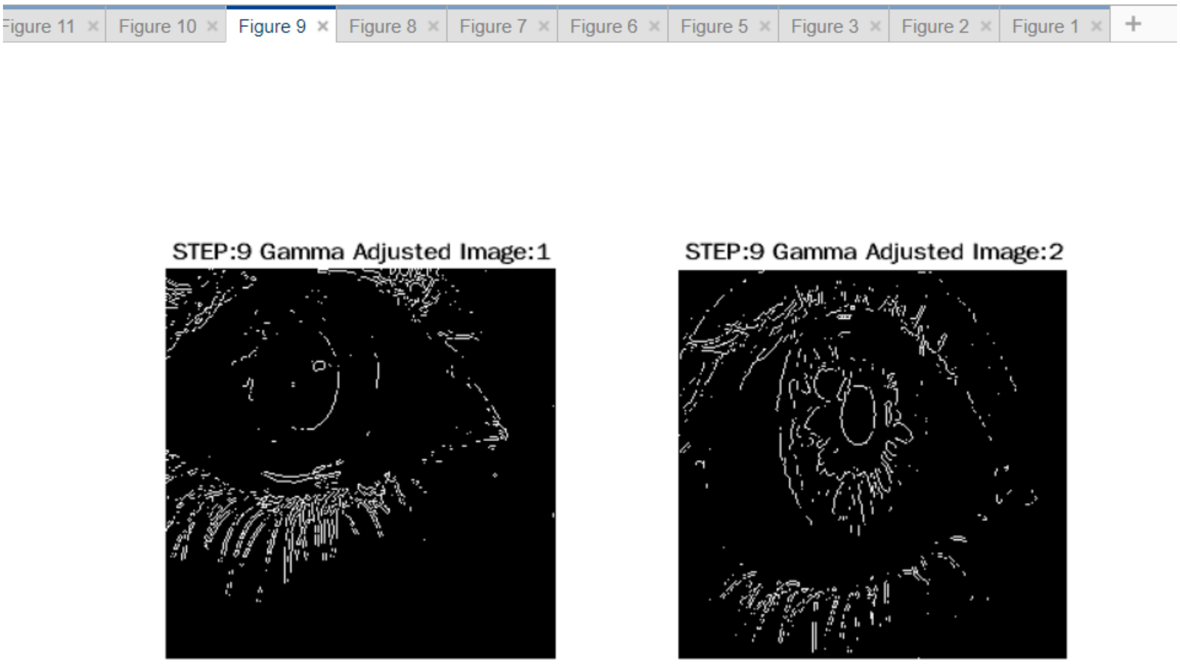


IMAGE SEGMENTATION:

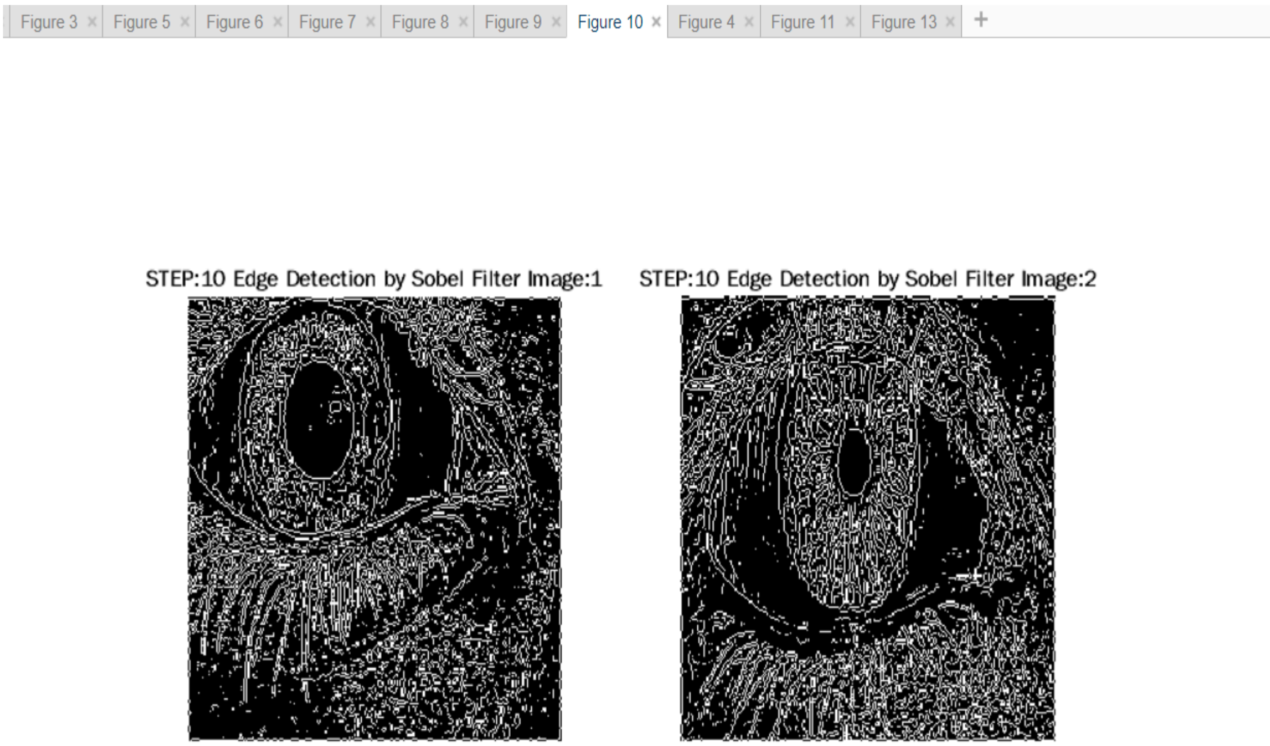
STEP:8 – CANNY EDGE DETECTION



STEP:9 – GAMMA CORRECTION (optional)

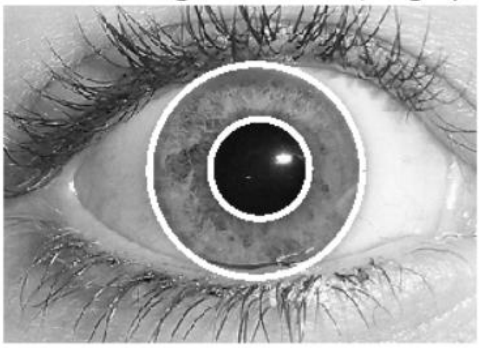


STEP:10 – EDGE DETECTION USING SOBEL FILTER (optional)

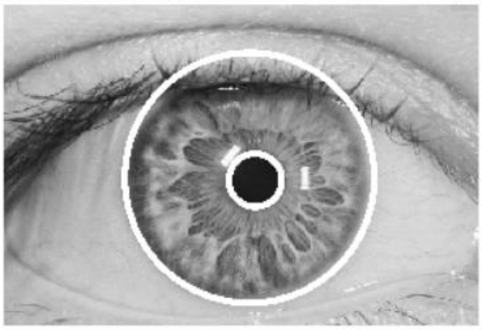


STEP:11 -HOUGH'S TRANSFORM TO DRAW IRIS AND PUPIL PERIMETER

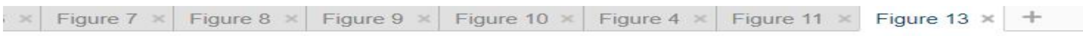
STEP:11 Houghs Transform (image1)



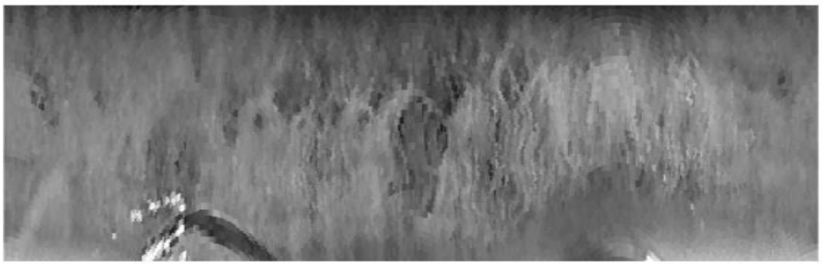
STEP:11 Houghs Transform (image2)



STEP:12 – STRETCHING NORMALISED IMAGES (AREA COVERED BY THE PERIMETER)



STEP:12 Normalised Image:1



STEP:12 Normalised Image:2

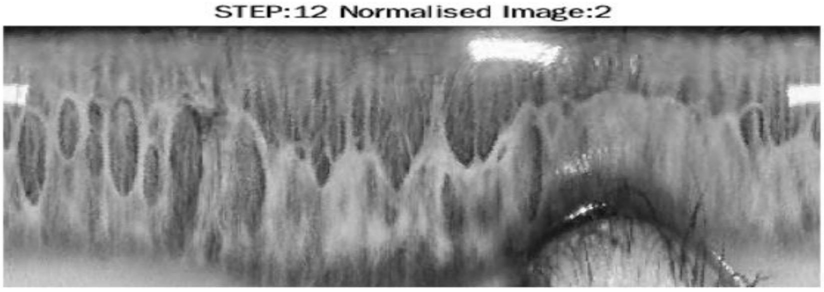


IMAGE1:

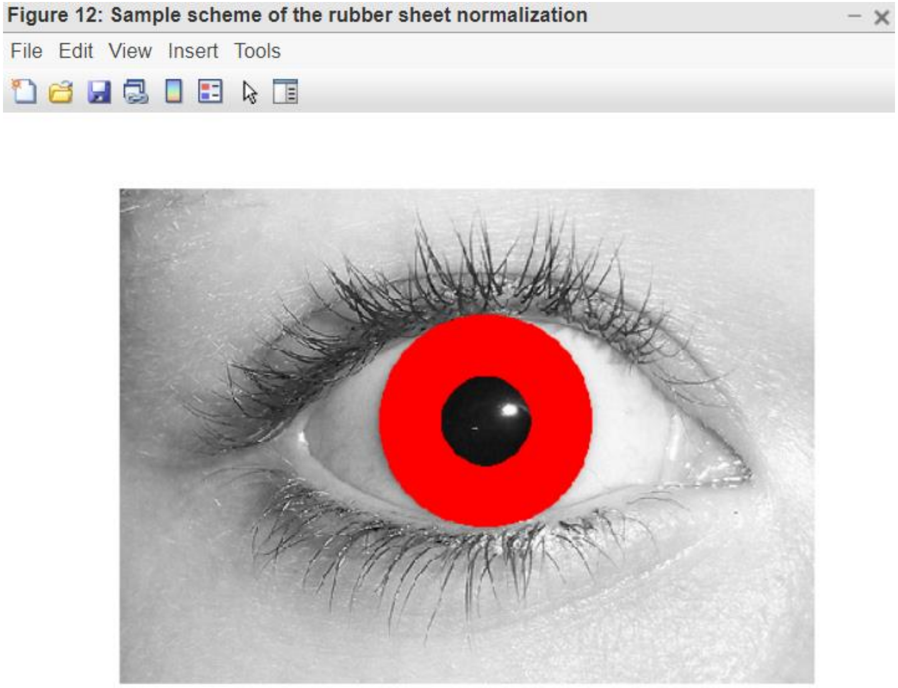
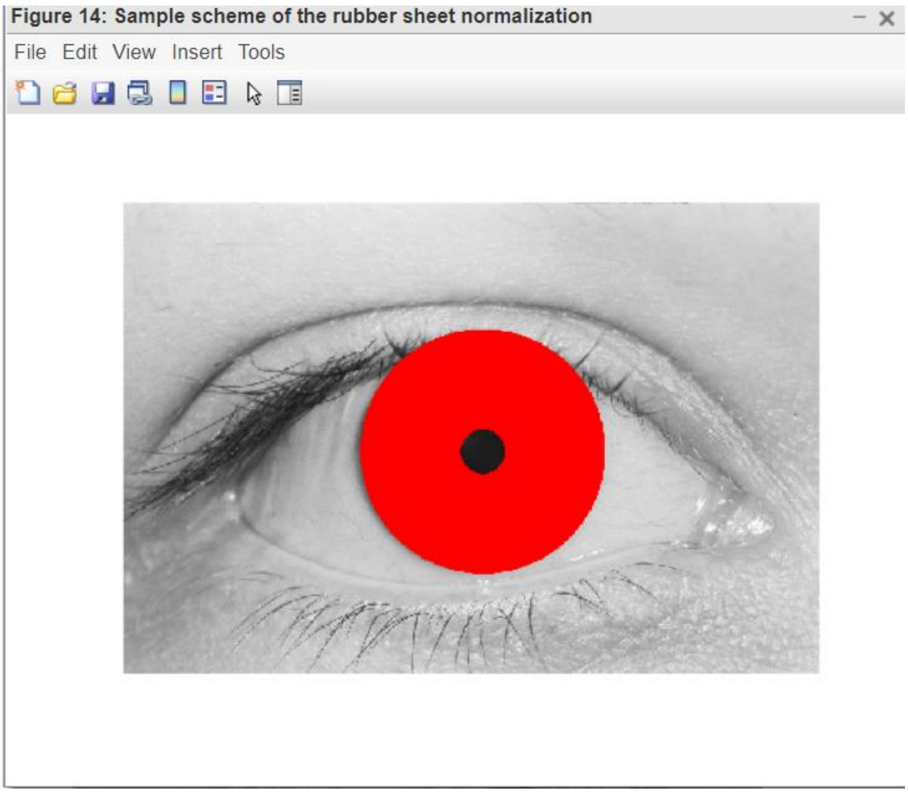
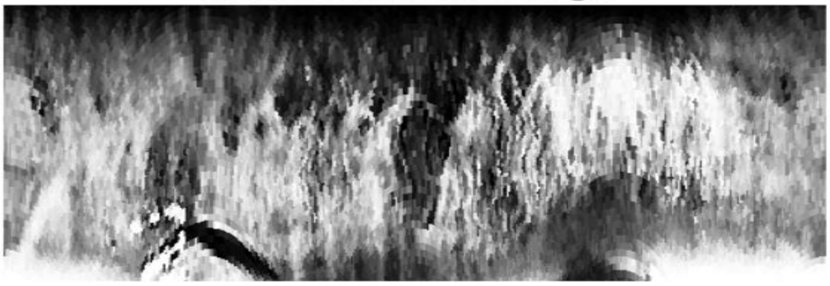


IMAGE2:

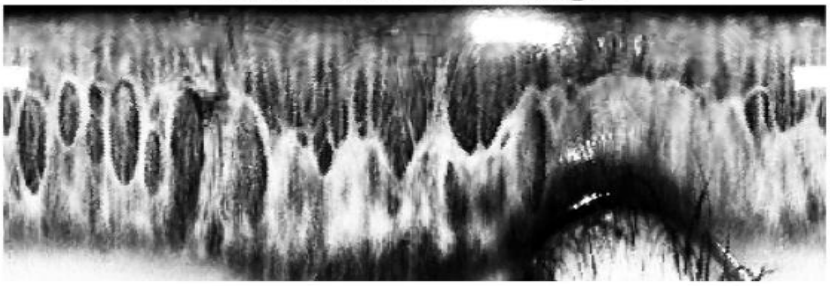


STEP:13-ENHANCEMENT OF NORMALISED IRIS IMAGES

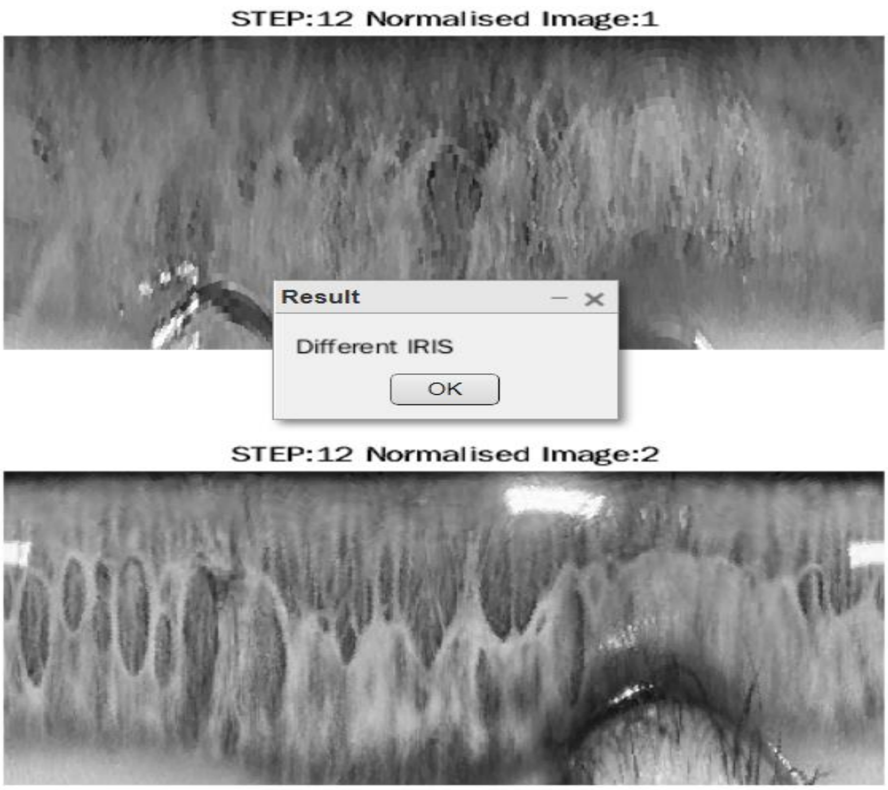
STEP:13-Enhanced iris image 1



STEP:13-enhanced iris image 2



STEP:14 AND 15-EXTRACTION AND MATCHING USING HAMMING DISTANCE:



STEP:15(B) (TO SHOW SAME IRIS RESULT, USING IDENTICAL IMAGES)