SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Permanently Affiliated to the University of Mumbai)

Department of Electronics & Telecommunication Engineering

Date:02/06/2022

A Mini Project on
"Iris Recognition using Matlab "

NAME	SAP ID
Aaryan Shah	60002190004
Disha Kunjadia	60002190035
Jeet Patel	60002190051
Neeti Adhia	60002190069

Index

Introduction:

Iris recognition is an automated method of biometric identification that uses mathematical pattern-recognition techniques on video images of one or both of the irises of an individual's eyes, whose complex patterns are unique, stable, and can be seen from some distance. Steps involved in Iris recognition are as follows:

1. **Image Acquisition:** Digital imaging or digital image acquisition is the creation of photographic images, such as of a physical scene or of the interior structure of an object. The term is often assumed to imply or include the processing, compression, storage, printing, and display of such images. Image Acquisition Toolbox provides functions and blocks that enable you to connect industrial and scientific cameras to MATLAB and Simulink. It includes a MATLAB app that lets you interactively detect and configure hardware properties.
2. **Converting into Gray Image**: Descriptive Statics has been used to find the maximum, minimum, standard deviation, mean and normally distribution of the data of all the variables of the study. Normal distribution of data shows the sensitivity of the variables towards the periodic changes and speculation. When the data is not normally distributed it means that the data is sensitive towards periodic changes and speculations which create the chances of arbitrage and the investors have the chance to earn above the normal profit.
3. **Subtract Gray Image:** Subtract one image from another or subtract constant from image. Z = imsubtract(X, Y) subtracts each element in array Y from the corresponding element in array X and returns the difference in the corresponding element of the output array Z. If X is an integer array, elements of the output that exceed the range of the integer type are truncated, and fractional values are rounded. Process Steps:

   1.Reading background image then foreground image.
   2. Converting both images from rgb to hsv.
   3. Bixoring background and foreground image.
   4. Converting rgb to grayscale.
   5. Reading rows and coloumns of image.
   6. Conveting it to binary image.

7. Applying median filter to remove noise.
8. Labeling boundary.
9. Removing noise.

4. **Histogram:** An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance. Image histograms are present on many modern digital cameras. Photographers can use them as an aid to show the distribution of tones captured, and whether image detail has been lost to blown-out highlights or blacked-out shadows. This is less useful when using a raw image format, as the dynamic range of the displayed image may only be an approximation to that in the raw file. The horizontal axis of the graph represents the tonal variations, while the vertical axis represents the number of pixels in that particular tone. The left side of the horizontal axis represents the black and dark areas, the middle represents medium grey and the right hand side represents light and pure white areas. The vertical axis represents the size of the area that is captured in each one of these zones. Thus, the histogram for a very dark image will have the majority of its data points on the left side and center of the graph. Conversely, the histogram for a very bright image with few dark areas and/or shadows will have most of its data points on the right side and center of the graph.

5. Image Segmentation: In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analysis. 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. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s).

6. Cropped Image: Cropping is the removal of the outer parts of an image to improve framing, accentuate subject matter or change aspect ratio. Depending on the application, this may be performed on a physical photograph, artwork or film footage, or achieved digitally using image editing software. The practice is common to the film, broadcasting, photographic, graphic design and printing industries. J = imcrop creates an interactive Crop Image tool associated with the image displayed in the current figure. With this syntax and the other interactive syntaxes, the Crop Image tool blocks the MATLAB® command line until you complete the operation. imcrop returns the cropped image, J. For more information about using the Crop Image tool, see Interactive Behavior.

7. Resized Image: Changing the size at which the image will print without changing the number of pixels in the image —instead the pixels are printed further apart or closer together. Resizing (scaling) an image will not affect screen display. Image Re sampling: Changing the number of pixels in the image.

8. Smoothing Image using Gaussian Filter: Gaussian smoothing filters are commonly used to reduce noise. Filter the image with isotropic Gaussian smoothing kernels of increasing standard deviations. Gaussian filters are generally isotropic, that is, they have the same standard deviation along both dimensions. An image can be filtered by an isotropic Gaussian filter by specifying a scalar value for sigma. Filter the image with anisotropic Gaussian smoothing kernels, imgaussfilt allows the Gaussian kernel to have different standard deviations along row and column dimensions. These are called axis-aligned anisotropic Gaussian filters. Specify a 2-element vector for sigma when using anisotropic filters. Low pass filtering (smoothing), is employed to remove high spatial frequency noise from a digital image.

9. Edge Detection Canny Filter: Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision.

Code:

```
%%%Image Aquisition Process:-DIFFERENT IRIS
clc; clear all; close all; warning off;
[FileName1,FilePath1] = uigetfile('*.jpg','Select the 1st 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('3.jpg');
subplot(1,2,1);
imshow(i);
title('STEP:1 Image Aquisition(Original Image:1)');
ii=imread('4.jpg');
subplot(1,2,2);
imshow(ii);
title('STEP:1 Image Aquisition(Original Image:2)');
figure();

% STEP:2 GRAY SCALE CONVERTION*********************************************

g =rgb2gray(i);
subplot(1,2,1);
imshow(g);
title('STEP:2 Converting into Gray Image:1');
gg=rgb2gray(ii);
subplot(1,2,2);
imshow(gg);
title('STEP:2 Converting into Gray Image:2');
figure();

% STEP:3 Subtraction of original image*************************************

k = imread('3.jpg');
v= rgb2gray(k);
v = imsubtract(v,60);
subplot(1,2,1);
imshow(v);
title('STEP:3 Subtracted Gray Image:1');
kk = imread('4.jpg');
vv =rgb2gray(kk);
vv = imsubtract(vv,60);
subplot(1,2,2);
imshow(vv);
title('STEP:3 Subtracted Gray Image:2');
figure();

% Find HISTOGRAM of the Image**********************************************
z=double(v);
subplot(1,2,1);
imhist(v);
axis off, axis tight;
title('STEP:4 Histogram of the Image:1');

zz=double(vv);
subplot(1,2,2);
```

```
imhist(vv);
axis off, axis tight;
title('STEP:4 Histogram of the Image:1');
figure();

% STEP:5 CROPED IMAGE*****************************************************

c=imcrop(g,[276 187 820-276 538-187]);
subplot(1,2,1);
imshow(c);
title('STEP:5 Croped Image:1');
cc=imcrop(gg,[137 102 616-137 352-102]);
subplot(1,2,2);
imshow(cc);
title('STEP:5 Croped 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 Using Gaussian Filter Smoothing Image:1 ');
ss= fspecial('gaussian',3);
ff = imfilter(rr,ss);
subplot(1,2,2);
imshow(ff,[]),title('STEP:7 Using 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 Edge Detection by Canny Filter Image:1');
ee=edge(ff,'canny');
subplot(1,2,2);
imshow(ee);
title('STEP:8 Edge Detection by Canny Filter Image:2');
figure();

%STEP:9 Sobel EDGE DETECTION***********************************************

S1=edge(f,'roberts');
subplot(1,2,1);
```

```
imshow(S1);
title('STEP:9 Edge Detection by Sobel Filter Image:1');
SS1=edge(ff,'roberts');
subplot(1,2,2);
imshow(SS1);
title('STEP:9 Edge Detection by Sobel Filter Image:2');
figure()

% STEP:11 GAMMA CORRECTION************************************************
%Adjusting the Gamma to 0.8

S=edge(f,'sobel');
u=double(S);
subplot(1,2,1);
y= imadjust(u,[],[],0.8);
imshow(y);
title('STEP:10 Gamma Adjusted Image:1');
SS=edge(ff,'sobel');
uu=double(SS);
subplot(1,2,2);
yy= imadjust(uu,[],[],0.8);
imshow(yy);
title('STEP:10 Gamma Adjusted Image:2');
figure();

% STEP:12 HISTERISIS THRASHOLD********************************************

mygrayimg = imread('3.jpg');
mygrayimg = imresize(rgb2gray(mygrayimg),[256 256]);
myfftimage = fft2(mygrayimg);
tmp = abs(myfftimage);
mylogimg = log(1+tmp);
[M,N] = size(myfftimage);
low = 62;
band1 = 15;
band2 = 60;
mylowpassmask = ones(M,N);
mybandpassmask = ones(M,N);
for u = 1:M
   for v = 1:N

       tmp = ((u-(M+1))/2)^2 +(v-(N+1)/2)^2;
       raddist = round((sqrt(tmp)));
       disp(raddist)

       if raddist > low
       mylowpassmask(u,v) = 0;
       end

       if raddist > band2 || raddist < band1;
       mybandpassmask(u,v) = 0;
       end
   end
end
f1 = fftshift(mylowpassmask);
f3 = fftshift(mybandpassmask);
resimage1 = myfftimage.*f1;
resimage3 = myfftimage.*f3;
```

```
% Display the low pass filtered image
r1 = abs(ifft2(resimage1));
subplot(1,2,1);
imshow(r1,[]),title('STEP:11 Hysteresis Thresholding of Image: 1');

% part 2*****
% Read the image, resize it to 256 x 256
% Convert it to grey image and display it

mygrayimg = imread('4.jpg');
mygrayimg = imresize(rgb2gray(mygrayimg),[256 256]);

% Finding FFT
% Using the command fft2() to get FFT of the image
% The log scale of FFT image is displayed

myfftimage = fft2(mygrayimg);

% Take logarithmic scale for display
tmp = abs(myfftimage);
mylogimg = log(1+tmp);

% Find size
[M,N] = size(myfftimage);

% Creating Filter array
% The cut off frequency 20 is used here
low = 62;
band1 = 15;
band2 = 60;

% creating ideal high pass filter mask
% Creating matrix of size equals original matrix
mylowpassmask = ones(M,N);
mybandpassmask = ones(M,N);

% Generate values for ideal high pass mask
for u = 1:M
   for v = 1:N

       tmp = ((u-(M+1))/2)^2 +(v-(N+1)/2)^2;
       raddist = round((sqrt(tmp)));
       disp(raddist)

       if raddist > low
       mylowpassmask(u,v) = 0;
       end

       if raddist > band2 || raddist < band1;
       mybandpassmask(u,v) = 0;
       end
   end
end

% Shift the spectrum to the centre
f1 = fftshift(mylowpassmask);
```

```
f3 = fftshift(mybandpassmask);

% Apply the filter H to the FFT of the Image
resimage1 = myfftimage.*f1;
resimage3 = myfftimage.*f3;


% Apply the Inverse FFT to the filtered image
% Display the low pass filtered image
r1 = abs(ifft2(resimage1));
subplot(1,2,2);
imshow(r1,[]),title('STEP:11 Hysteresis Thresholding of Image: 2');
figure();

% STEP:12 HUGH TRANSFORM***************************************************

Hu=imread('hug3.jpg');
subplot(1,2,1);
imshow(Hu);
title('STEP:12 After Hugh Transformation Image:1')
Hu1=imread('hug4.jpg');
subplot(1,2,2);
imshow(Hu1);
title('STEP:12 After Hugh Transformation Image:2')
figure();


% STEP:13 NORMALIZATION****************************************************

n=imread('normal3.jpg');
subplot(2,1,1);
imshow(n);
title('STEP:13 Normalized Image:1')
nn=imread('normal4.jpg');
subplot(2,1,2);
imshow(nn);
title('STEP:13 Normalized Image:2')
figure();

% STEP:14 MATCHING*********************************************************

N=im2double(f);
subplot(1,2,1);
imshow(N);
title('STEP:14 IMAGE 1');
NN=im2double(ff);
subplot(1,2,2);
imshow(NN);
title('STEP:14 IMAGE 2');

% STEP:15 MATCHING*********************************************************

w = msgbox('DIFFERENT IRIS','Result');
```

Results:

STEP:1 Image Aquisition(Original Image:1)

STEP:1 Image Aquisition(Original Image:2)

STEP:2 Converting into Gray Image:1

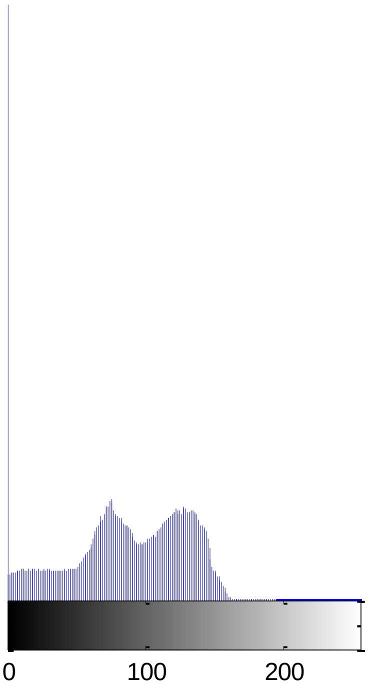
STEP:2 Converting into Gray Image:2

STEP:3 Subtracted Gray Image:1

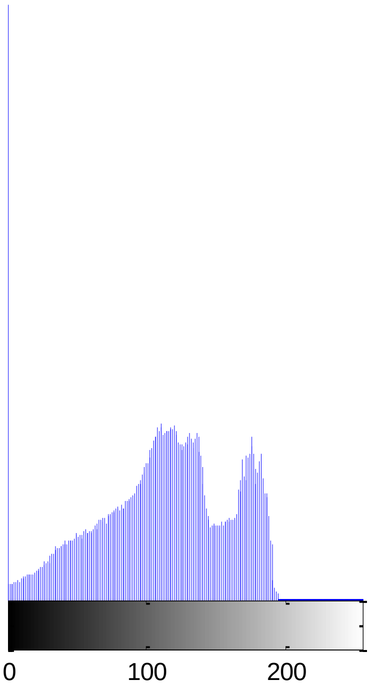
STEP:3 Subtracted Gray Image:2

Department of Electronics & Telecommunication Engineering

STEP:4 Histogram of the Image:1



STEP:4 Histogram of the Image:1



STEP:5 Croped Image:1

STEP:5 Croped Image:2

STEP:6 Resized Image:1

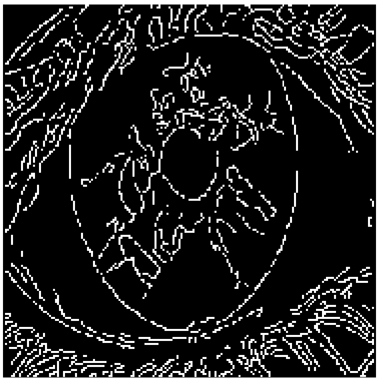
STEP:6 Resized Image:2

STEP:7 Using Gaussian Filter Smoothing Image:1

STEP:7 Using Gaussian Filter Smoothing Image:2

STEP:8 Edge Detection by Canny Filter Image:1

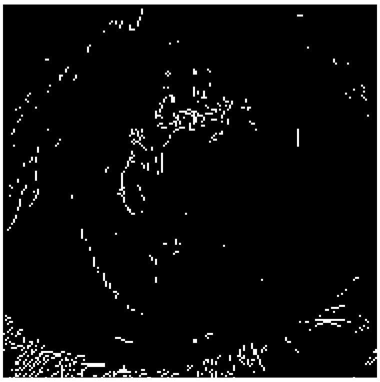
STEP:8 Edge Detection by Canny Filter Image:2



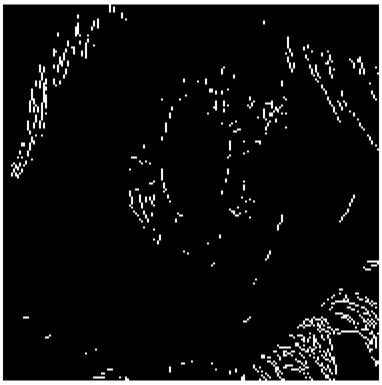
STEP:9 Edge Detection by Sobel Filter Image:1

STEP:9 Edge Detection by Sobel Filter Image:2

STEP:10 Gamma Adjusted Image:1



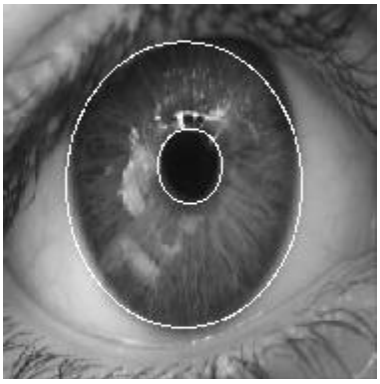
STEP:10 Gamma Adjusted Image:2



STEP:11 Hysteresis Thresholding of Image: 1

STEP:11 Hysteresis Thresholding of Image: 2

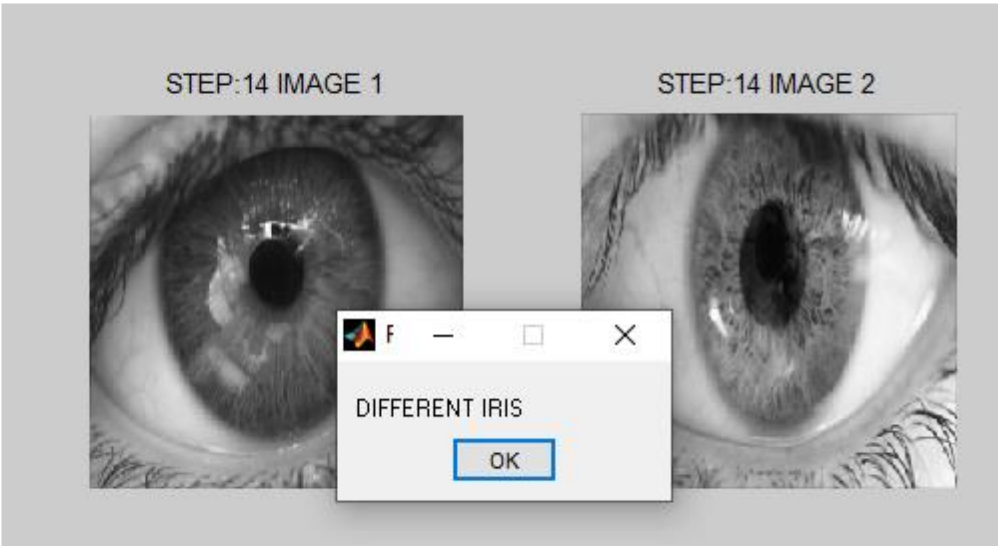
STEP:12 After Hugh Transformation Image:1

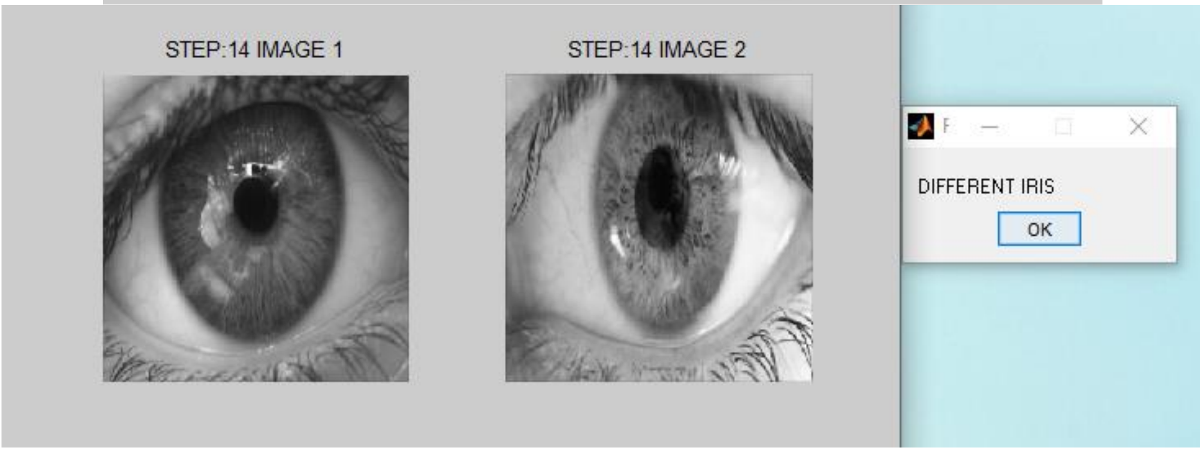


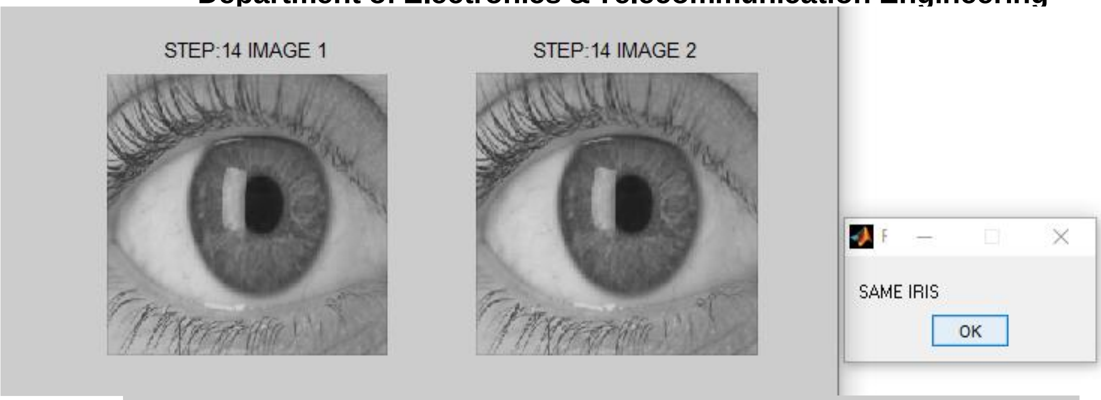
STEP:12 After Hugh Transformation Image:2

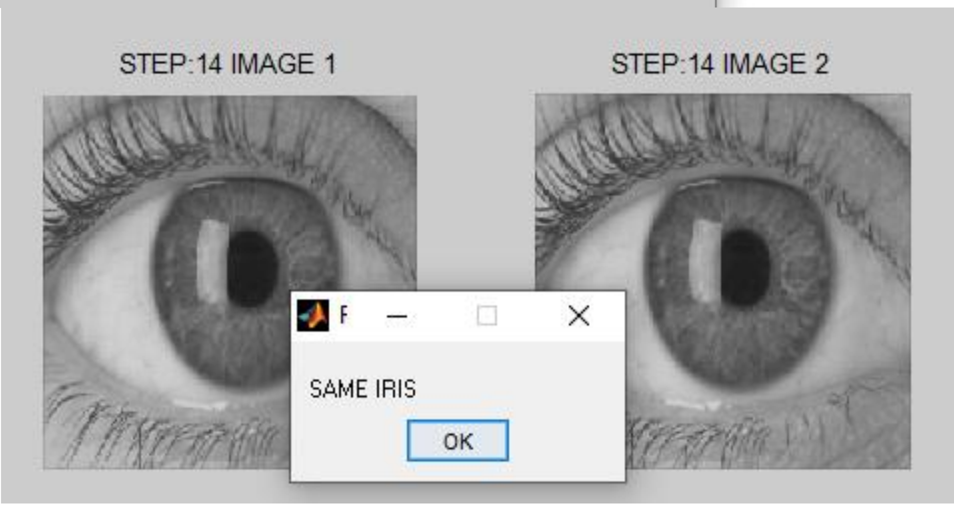
STEP:13 Normalized Image:1

STEP:13 Normalized Image:2






STEP:14 IMAGE 1
STEP:14 IMAGE 2
SAME IRIS
OK


STEP:14 IMAGE 1
STEP:14 IMAGE 2
SAME IRIS
OK

Conclusion:

Here we have achieved the goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analysis. Edge detection is an image processing technique for finding the boundaries of objects within images and works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision The process of recognizing a person by examining the random pattern of the iris is achieved here. The automatic iris recognition system is quite young. The eye muscle that controls the amount of light that enters the eye by regulating the size of the pupil. We have successfully found the colored part of the eye that changes color according on the amount of melatonin pigment in the eye and using filters detected if it is the same iris or a different iris.

How to improve the model / Future Enhancements?

1) In order to increase both accuracy and robustness; a multimodal biometric system could be used.
2) This confusion may be a combination of iris and fingerprint biometrics.
3) This allows the integration of two or more types of biometric recognition and verification systems in order to meet stringent performance requirements.
4) Such systems are expected to be more reliable due to the presence of multiple independent pieces of evidence.
5) These systems are also able to meet the strict performance requirements imposed by various applications

References:

[1].Mohd Nasrullah Kazmi, MohdFaiz under the Guidance of Prof. Bandanawaz "Iris Recognition System using MATLAB" PDF file from college ANJUMAN I-ISLAM KALSEKHAR TECHNICAL CAMPUS by the students of Department of Electronics and Telecommunication Enginnering.

[2].John Daugman "How Iris Recognition Works" University of Cambridge CB2 3QG, U.K.

[3]. C.H. Daouk, L.A. El-Esber, F.D. Kammoun and M.A. Al Alaoui, "Iris Recognition" preceding of IEEE International Symposium on Signal Processing and Information Technology,2002,pp. 558-562.

[4]. Bhawna chouhan, shailja Shukla. "Iris Recognition System Using Canny Edge Detection for Biometric Identification", in the International Journal of Engineering Sciences and Technology, ISSN: 0975-5462 Vol. 3 No.1 Jan 2011.

[5]. B.Sabrigiiri, T.Karthikeyan "Acquisition of Iris Images, Iris Localization, Normalization, and Quality Enhancement for Personal Identification" in the International Journal of Engineering Trends & Technology in Computer Science, ISSN 2278-6856,pp 274-275, Volume 1, Issue 2, July-August 2012

[6]. HaneneGuesmj et al. "Novel Iris Segmentation Model" in the preceding of International Conference on Multimedia Computing and System, third edition 2012.