Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084

CS 559 ASSIGNMENT 2

Design and evaluation of Iris Recognition System

REPORT CONTENTS.

Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084

Iris Recognition:

The iris recognition system consists of an auto-matic segmentation system that is based on Hough Transform. The code used in this is able to localise the circular iris and pupil region, occuliding eyelids and eyelashes and reflections.

① The extracted image is then normalised into a rectangular block with constant dimensions.

② Finally, the phase data from 1D Log-Gabor filters was extracted and quantised to four levels to encode the unique pattern of the iris to a bit-wise template

③ Hamming distance was then employed for classification of iris templates and compared.

a) Three iris data set were given to us and images taken from same subjects both eyes are stored in same subfolder with right/left eye indications.

Among the data set given we have probes and gallery.

P.T.O

PROBE:-

A fresh biometric sample to be recognized is called as probe or query.

GALLERY:-

The set of enrolled identifiers is called the gallery or watch list.

Here. Database LG2200 2008 are used as gallery
LG4000, LG2200 - used as probes.

In our experiment we have taken 25 cases, and are probes are from both the database given.

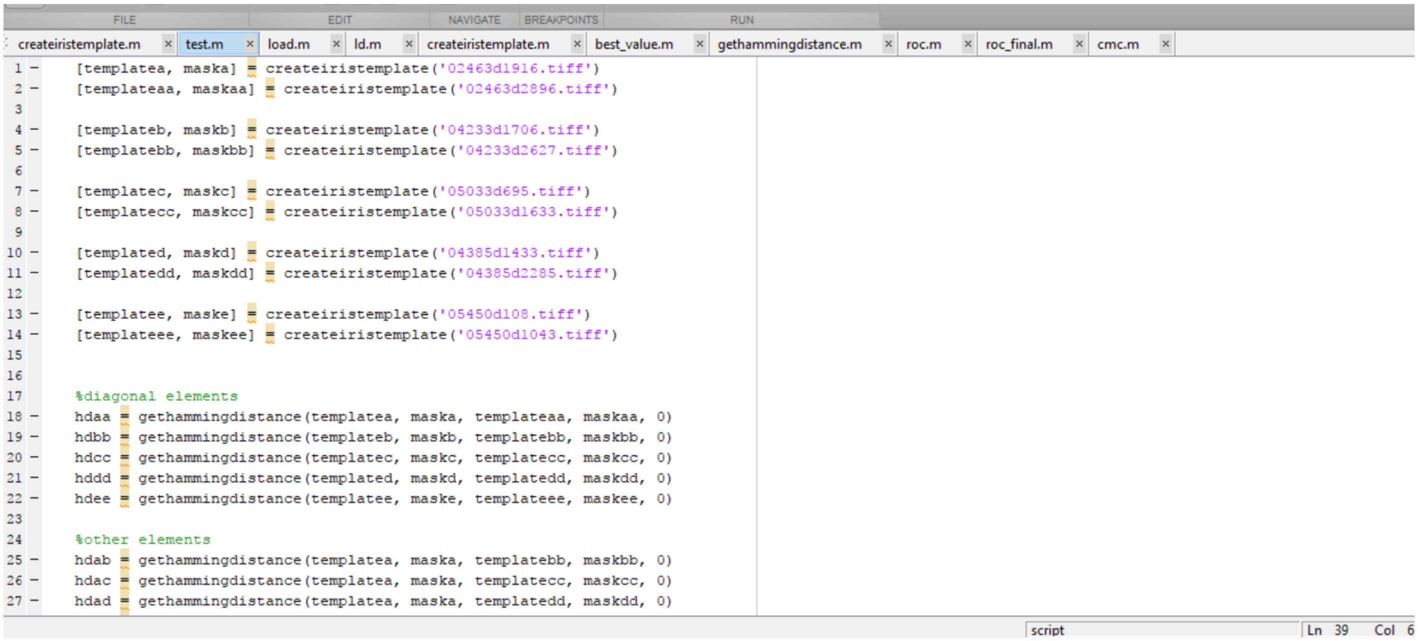
b) DESIGN:-

-> we first created the database from the given data set for probes and query.

-> We called the function createiristemplate.m which creates templates for the image, where noise, segmentation, matching, normalisation, encoding templates are created in the diagnostic folder.

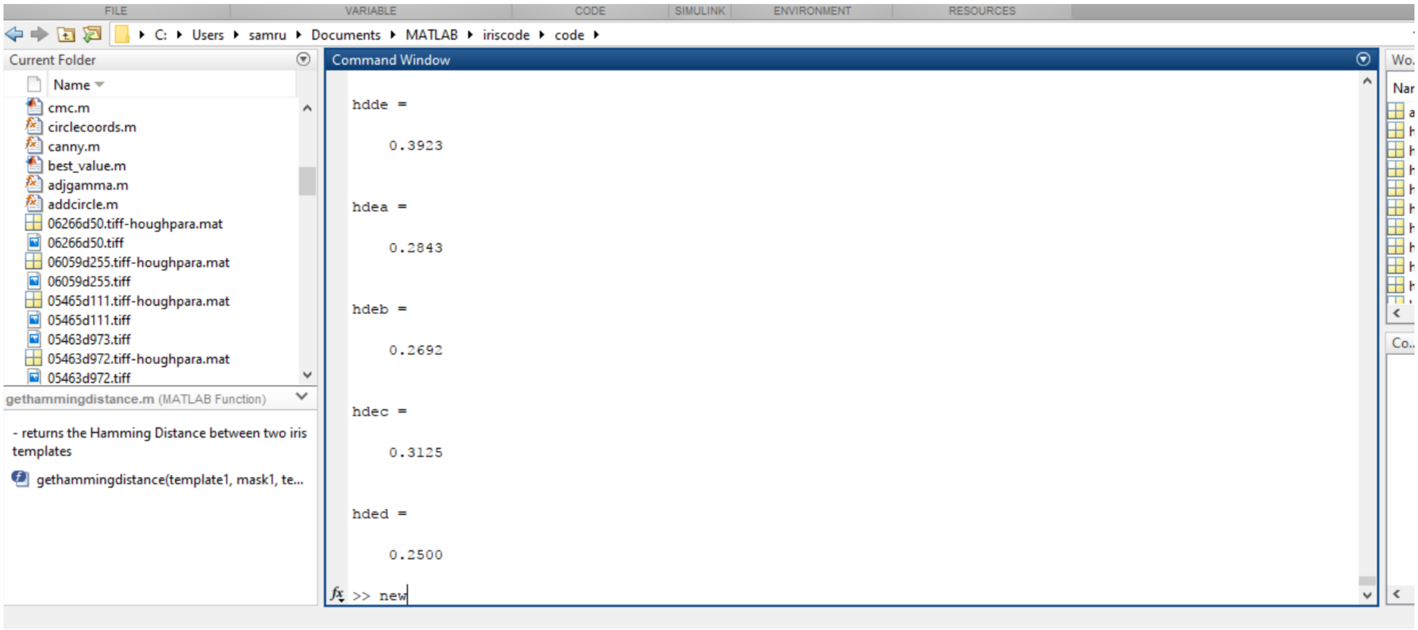
Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084

Code to load data and get hamming distance - test.m



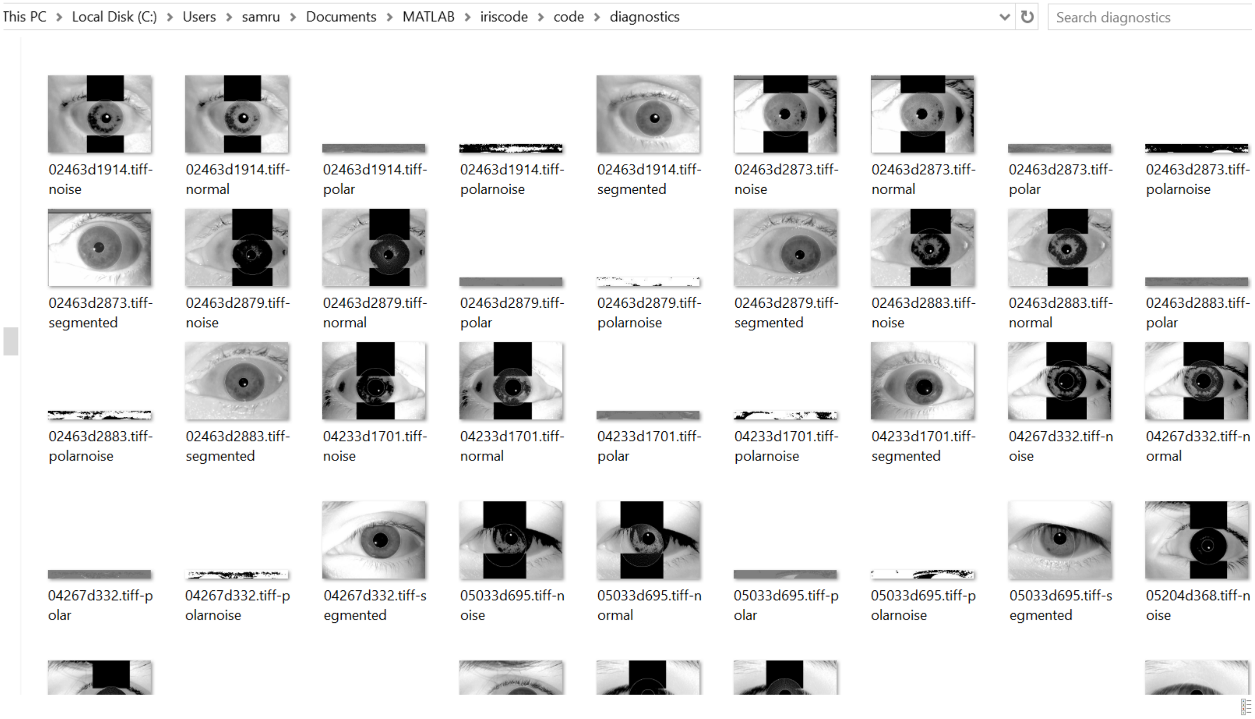
```
[templatea, maska] = createiristemplate('02463d1916.tiff')
[templateaa, maskaa] = createiristemplate('02463d2896.tiff')

[templateb, maskb] = createiristemplate('04233d1706.tiff')
[templatebb, maskbb] = createiristemplate('04233d2627.tiff')

[templatec, maskc] = createiristemplate('05033d695.tiff')
[templatecc, maskcc] = createiristemplate('05033d1633.tiff')

[templated, maskd] = createiristemplate('04385d1433.tiff')
[templatedd, maskdd] = createiristemplate('04385d2285.tiff')

[templatee, maske] = createiristemplate('05450d108.tiff')
[templateee, maskee] = createiristemplate('05450d1043.tiff')


%diagonal elements
hdaa = gethammingdistance(templatea, maska, templateaa, maskaa, 0)
hdbb = gethammingdistance(templateb, maskb, templatebb, maskbb, 0)
hdcc = gethammingdistance(templatec, maskc, templatecc, maskcc, 0)
hddd = gethammingdistance(templated, maskd, templatedd, maskdd, 0)
hdee = gethammingdistance(templatee, maske, templateee, maskee, 0)

%other elements
hdab = gethammingdistance(templatea, maska, templatebb, maskbb, 0)
hdac = gethammingdistance(templatea, maska, templatecc, maskcc, 0)
hdad = gethammingdistance(templatea, maska, templatedd, maskdd, 0)
```

Output:-


FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ samru ▸ Documents ▸ MATLAB ▸ iriscode ▸ code ▸
Current Folder
Name
cmc.m
circlecoords.m
canny.m
best_value.m
adjgamma.m
addcircle.m
06266d50.tiff-houghpara.mat
06266d50.tiff
06059d255.tiff-houghpara.mat
06059d255.tiff
05465d111.tiff-houghpara.mat
05465d111.tiff
05463d973.tiff
05463d972.tiff-houghpara.mat
05463d972.tiff
gethammingdistance.m (MATLAB Function)
- returns the Hamming Distance between two iris templates
gethammingdistance(template1, mask1, te...
Command Window
Wo
Nar
Co..

```
hdde =

    0.3923


hdea =

    0.2843


hdeb =

    0.2692


hdec =

    0.3125


hded =

    0.2500

>> new
```

Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084



This PC > Local Disk (C:) > Users > samru > Documents > MATLAB > iriscode > code > diagnostics

Search diagnostics

02463d1914.tiff-noise
02463d1914.tiff-normal
02463d1914.tiff-polar
02463d1914.tiff-polarnoise
02463d1914.tiff-segmented
02463d2873.tiff-noise
02463d2873.tiff-normal
02463d2873.tiff-polar
02463d2873.tiff-polarnoise
02463d2873.tiff-segmented
02463d2879.tiff-noise
02463d2879.tiff-normal
02463d2879.tiff-polar
02463d2879.tiff-polarnoise
02463d2879.tiff-segmented
02463d2883.tiff-noise
02463d2883.tiff-normal
02463d2883.tiff-polar
02463d2883.tiff-polarnoise
02463d2883.tiff-segmented
04233d1701.tiff-noise
04233d1701.tiff-normal
04233d1701.tiff-polar
04233d1701.tiff-polarnoise
04233d1701.tiff-segmented
04267d332.tiff-noise
04267d332.tiff-normal
04267d332.tiff-polar
04267d332.tiff-polarnoise
04267d332.tiff-segmented
05033d695.tiff-noise
05033d695.tiff-normal
05033d695.tiff-polar
05033d695.tiff-polarnoise
05033d695.tiff-segmented
05204d368.tiff-noise

Matrix for 25 test cases (Gallery to Probe)-

	a	b	c	d	e
a	0.200	0.4505	0.4388	0.4113	0.4446
b	0.3253	0.010	0.3596	0.2979	0.3769
c	0.3492	0.2949	0.121	0.3866	0.4269
d	0.3152	0.2550	0.3588	0.2920	0.3923
e	0.2895	0.2692	0.3125	0.2500	0.3333

~~2~~ 4

→ Then we compare the templates of the image by calling the ~~Ha~~ gethammingdistance-m.

→ Here Matching is done with hamming distance.

Hamming distance:

It gives the measure of how many bits are same between two bit patterns of the templates. Using the Hamming distance of two bit patterns, a decision can be made as to whether the two patterns were generated from different irises or from the same one.

$$HD = \frac{\text{Sum of disagreeing bits (exclusive OR)}}{N \text{ (total bits)}}$$

$0 < HD < 1$; 0.5 - random agreement
0.0 - perfect, but not possible due to noise and other factors

c) → Impostor & Genuine distributions.

The same-subfolder images from the probe and the gallery should give the closest identification value along diagnol. With this the matrix is drawn and distributions are plotted

Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084

E

Probe

Gallery		A	B	C	D	E
	F	0·2	0·45	0·43	0·41	0·44
	G	0·32	0·01	035	0·29	0·37
	H	0·34	0·29	0·12	0·38	0·42
	I	0·31.	0·25	0·35	0·29	0·39.
	J	0·28	0·26	0·37	0·25	0·33

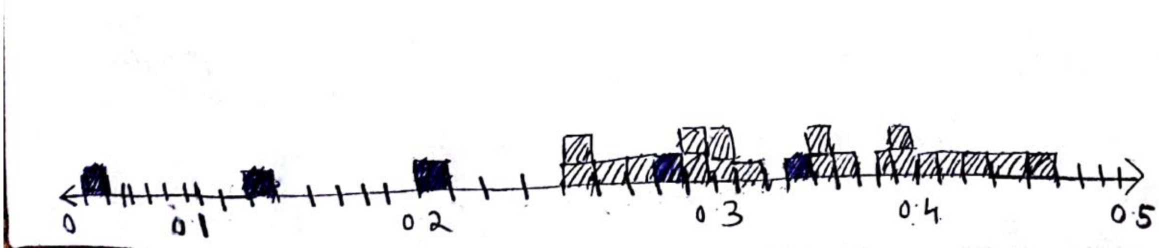
ii)

The matrix has the hamming distance calculated for the probe and gallery. Since the diagnol probe / gallery are from the same subset, the diagnol has less HD (ie) they are the genuine scores-

The genuine & imposter distribution:-

▩ - Genuine

▨ → imposter



Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084

③

d) Now to plot the ROC curve we vary the threshold (t) and plot the graph between FMR and TMR

$$FMR = \frac{\text{No. of false Match.}}{\text{No. of false identity claims}}$$

$$TMR = 1 - FNMR = 1 - \frac{\text{No. of false non-match results}}{\text{No. of True identity claims}}$$

Varying threshold.

① $t = 0.2.$

$FMR = 0$

$TMR = 0.6$

② $t = 0.29.$

$FMR = 0.2$

$FMR = 0.2.$

③ $t = 0.18$

$FMR = 0$

$TMR = 0.2$

④ $t = 0.25$

$FMR = 0.$

$TMR = 0.6$

⑤ $t = 0.32$

$FMR = 0.4.$

$TMR = 0.8$

⑥ $t = 0.4$

$FMR = 0.75$

$TMR = 0$

⑦ $t = 0.35.$

$FMR = 0.45$

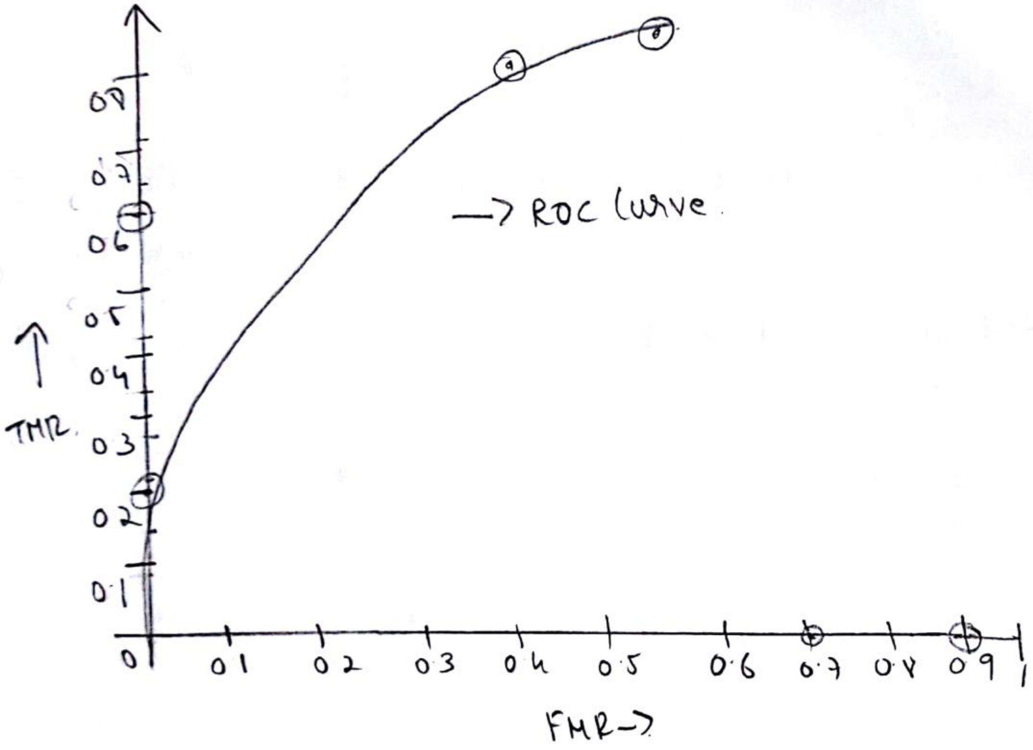
$TMR = 0.7$

⑧ $t = 0.45$

$FMR = 0.9$

$TMR = 0.$

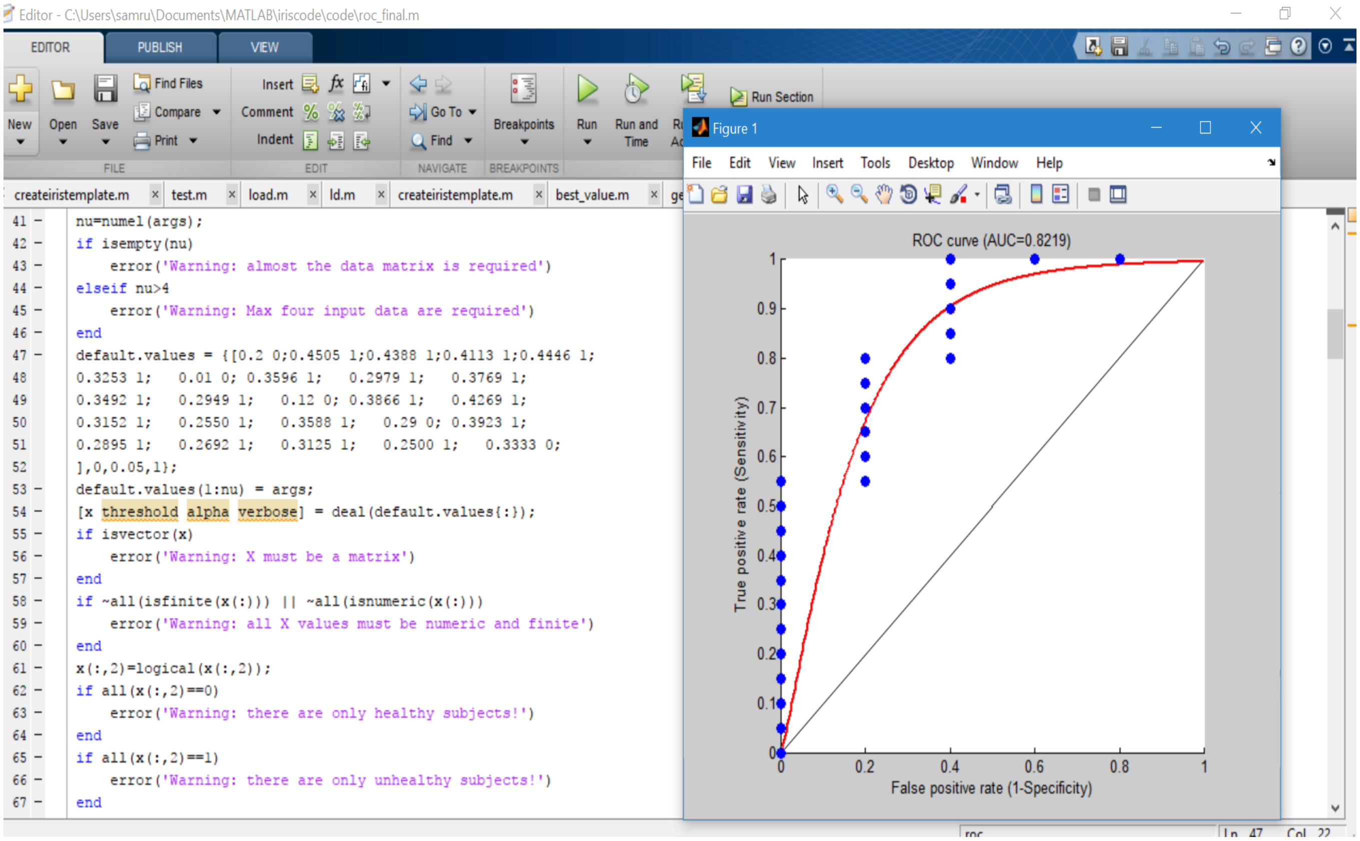
7



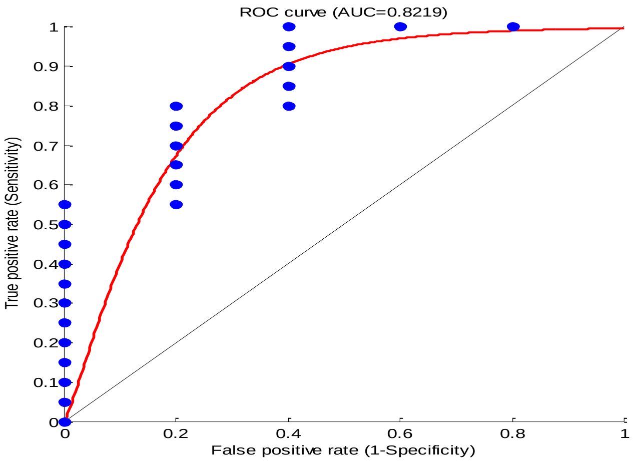
→ We have also coded in matlab to find the ROC curve

→ ROC curve thus found is pasted in screenshot folder.

Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084



```
nu=numel(args);
if isempty(nu)
    error('Warning: almost the data matrix is required')
elseif nu>4
    error('Warning: Max four input data are required')
end
default.values = {[0.2 0;0.4505 1;0.4388 1;0.4113 1;0.4446 1;
0.3253 1;   0.01 0; 0.3596 1;   0.2979 1;   0.3769 1;
0.3492 1;   0.2949 1;   0.12 0; 0.3866 1;   0.4269 1;
0.3152 1;   0.2550 1;   0.3588 1;   0.29 0; 0.3923 1;
0.2895 1;   0.2692 1;   0.3125 1;   0.2500 1;   0.3333 0;
],0,0.05,1};
default.values(1:nu) = args;
[x threshold alpha verbose] = deal(default.values{:});
if isvector(x)
    error('Warning: X must be a matrix')
end
if ~all(isfinite(x(:))) || ~all(isnumeric(x(:)))
    error('Warning: all X values must be numeric and finite')
end
x(:,2)=logical(x(:,2));
if all(x(:,2)==0)
    error('Warning: there are only healthy subjects!')
end
if all(x(:,2)==1)
    error('Warning: there are only unhealthy subjects!')
end
```



Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084

e) CMC curve.

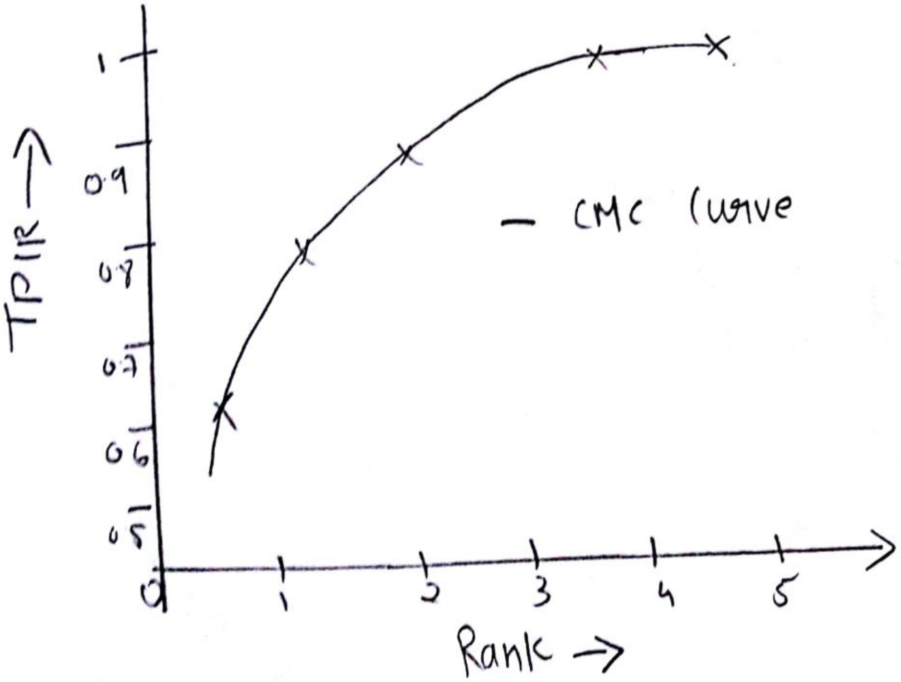
→ In CMC curve each biometric sample is compared to all the gallery samples.

→ The resulting scores are sorted and ranked.

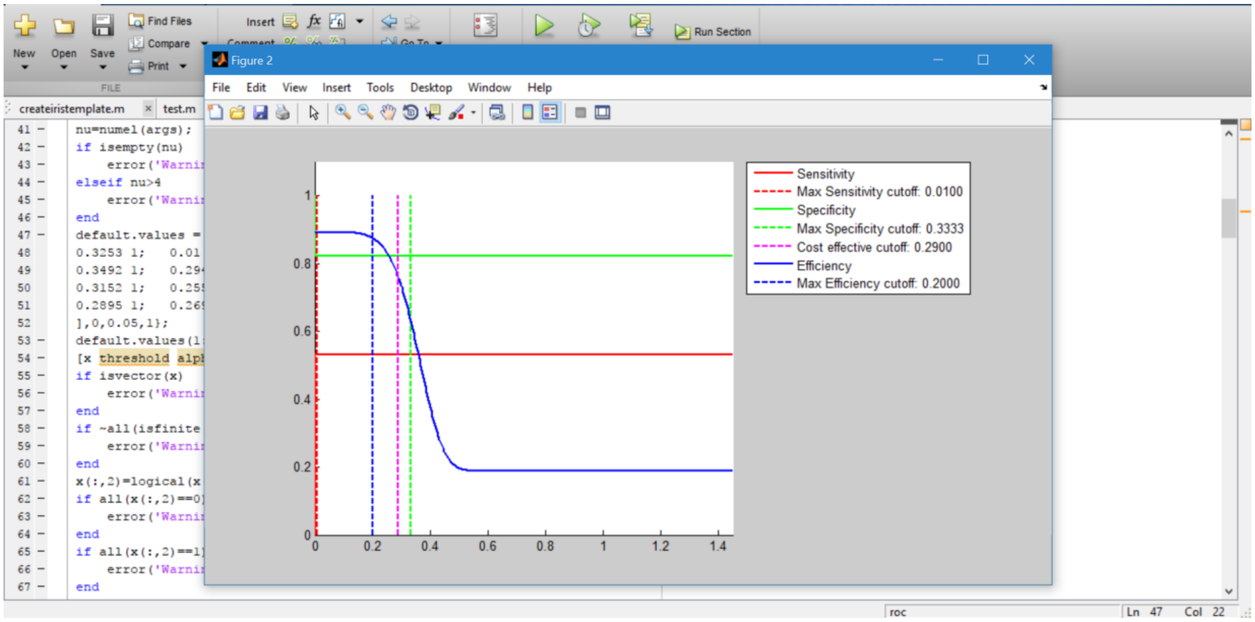
→ TPIR: Probability of observing the correct identity with top k ranks.

→ R_i is the rank number of the observation x_i and i indicates the serial number in the range of ascendind data.

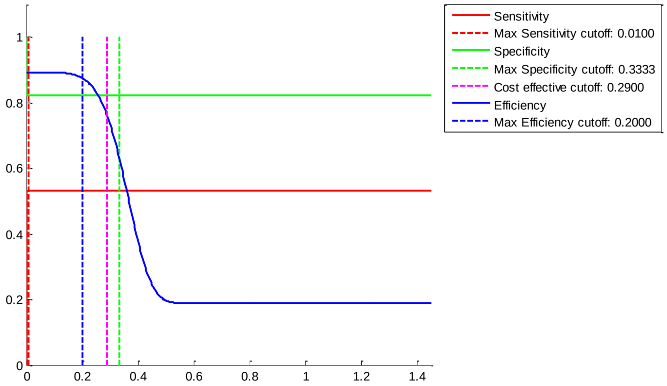
$$P_c = R_i/(N+1)$$



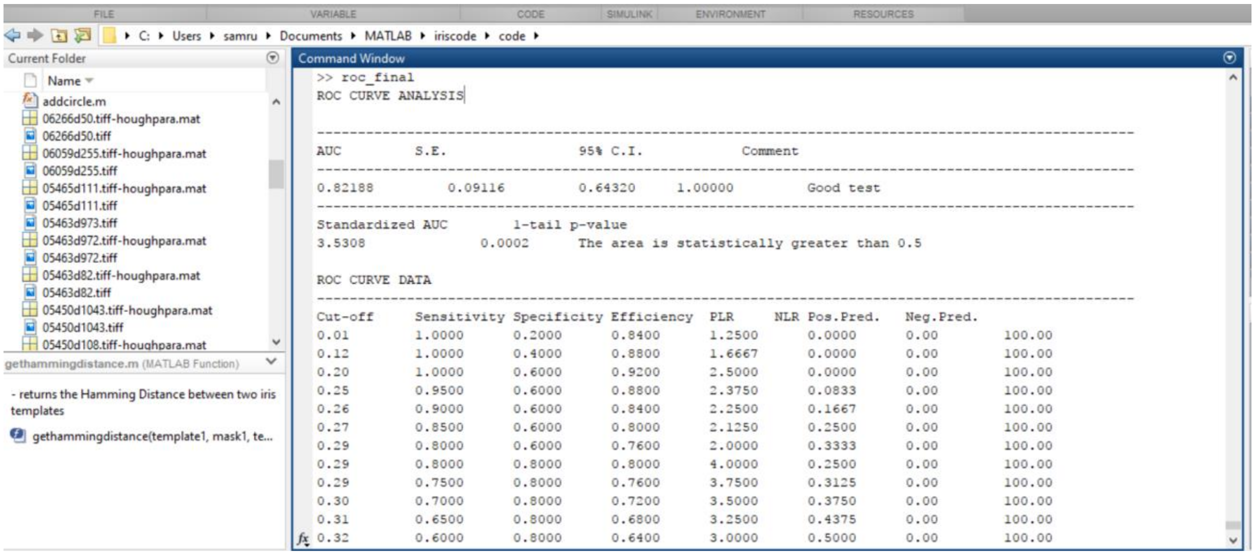
Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084



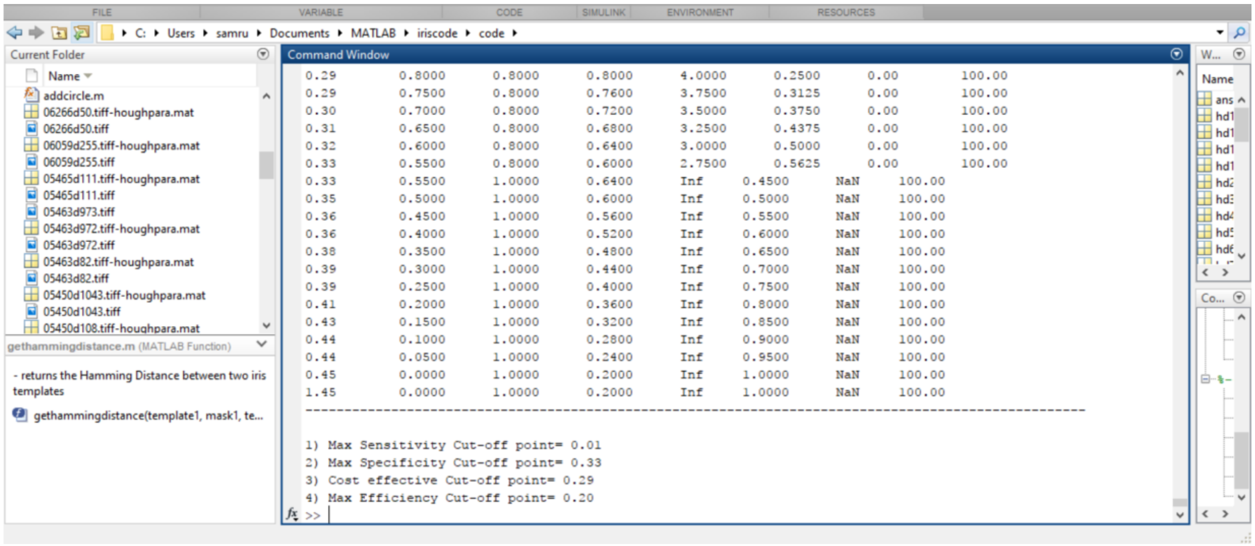
```
nu=numel(args);
if isempty(nu)
    error('Warni
elseif nu>4
    error('Warni
end
default.values =
0.3253 1;   0.01
0.3492 1;   0.29
0.3152 1;   0.25
0.2895 1;   0.26
],0,0.05,1};
default.values(1
[x threshold alp
if isvector(x)
    error('Warni
end
if ~all(isfinite
    error('Warni
end
x(:,2)=logical(x
if all(x(:,2)==0)
    error('Warni
end
if all(x(:,2)==1)
    error('Warni
end
```



Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084


FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ samru ▸ Documents ▸ MATLAB ▸ iriscode ▸ code ▸
Current Folder
Name
addcircle.m
06266d50.tiff-houghpara.mat
06266d50.tiff
06059d255.tiff-houghpara.mat
06059d255.tiff
05465d111.tiff-houghpara.mat
05465d111.tiff
05463d973.tiff
05463d972.tiff-houghpara.mat
05463d972.tiff
05463d82.tiff-houghpara.mat
05463d82.tiff
05450d1043.tiff-houghpara.mat
05450d1043.tiff
05450d108.tiff-houghpara.mat
gethammingdistance.m (MATLAB Function)
- returns the Hamming Distance between two iris templates
gethammingdistance(template1, mask1, te...
Command Window

```
>> roc_final
ROC CURVE ANALYSIS

--------------------------------------------------------------------------------------------------
AUC          S.E.                 95% C.I.            Comment
--------------------------------------------------------------------------------------------------
0.82188         0.09116         0.64320     1.00000         Good test
--------------------------------------------------------------------------------------------------
Standardized AUC        1-tail p-value
3.5308              0.0002      The area is statistically greater than 0.5

ROC CURVE DATA
--------------------------------------------------------------------------------------------------
Cut-off      Sensitivity Specificity Efficiency  PLR     NLR Pos.Pred.   Neg.Pred.
0.01         1.0000      0.2000      0.8400      1.2500      0.0000      0.00        100.00
0.12         1.0000      0.4000      0.8800      1.6667      0.0000      0.00        100.00
0.20         1.0000      0.6000      0.9200      2.5000      0.0000      0.00        100.00
0.25         0.9500      0.6000      0.8800      2.3750      0.0833      0.00        100.00
0.26         0.9000      0.6000      0.8400      2.2500      0.1667      0.00        100.00
0.27         0.8500      0.6000      0.8000      2.1250      0.2500      0.00        100.00
0.29         0.8000      0.6000      0.7600      2.0000      0.3333      0.00        100.00
0.29         0.8000      0.8000      0.8000      4.0000      0.2500      0.00        100.00
0.29         0.7500      0.8000      0.7600      3.7500      0.3125      0.00        100.00
0.30         0.7000      0.8000      0.7200      3.5000      0.3750      0.00        100.00
0.31         0.6500      0.8000      0.6800      3.2500      0.4375      0.00        100.00
0.32         0.6000      0.8000      0.6400      3.0000      0.5000      0.00        100.00
```


FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ samru ▸ Documents ▸ MATLAB ▸ iriscode ▸ code ▸
Current Folder
Name
addcircle.m
06266d50.tiff-houghpara.mat
06266d50.tiff
06059d255.tiff-houghpara.mat
06059d255.tiff
05465d111.tiff-houghpara.mat
05465d111.tiff
05463d973.tiff
05463d972.tiff-houghpara.mat
05463d972.tiff
05463d82.tiff-houghpara.mat
05463d82.tiff
05450d1043.tiff-houghpara.mat
05450d1043.tiff
05450d108.tiff-houghpara.mat
gethammingdistance.m (MATLAB Function)
- returns the Hamming Distance between two iris templates
gethammingdistance(template1, mask1, te...
Command Window
W...
Name
ans
hd1
hd1
hd1
hd1
hd2
hd3
hd4
hd5
hd6
Co...

```
0.29         0.8000      0.8000      0.8000      4.0000      0.2500      0.00        100.00
0.29         0.7500      0.8000      0.7600      3.7500      0.3125      0.00        100.00
0.30         0.7000      0.8000      0.7200      3.5000      0.3750      0.00        100.00
0.31         0.6500      0.8000      0.6800      3.2500      0.4375      0.00        100.00
0.32         0.6000      0.8000      0.6400      3.0000      0.5000      0.00        100.00
0.33         0.5500      0.8000      0.6000      2.7500      0.5625      0.00        100.00
0.33         0.5500      1.0000      0.6400      Inf     0.4500      NaN     100.00
0.35         0.5000      1.0000      0.6000      Inf     0.5000      NaN     100.00
0.36         0.4500      1.0000      0.5600      Inf     0.5500      NaN     100.00
0.36         0.4000      1.0000      0.5200      Inf     0.6000      NaN     100.00
0.38         0.3500      1.0000      0.4800      Inf     0.6500      NaN     100.00
0.39         0.3000      1.0000      0.4400      Inf     0.7000      NaN     100.00
0.39         0.2500      1.0000      0.4000      Inf     0.7500      NaN     100.00
0.41         0.2000      1.0000      0.3600      Inf     0.8000      NaN     100.00
0.43         0.1500      1.0000      0.3200      Inf     0.8500      NaN     100.00
0.44         0.1000      1.0000      0.2800      Inf     0.9000      NaN     100.00
0.44         0.0500      1.0000      0.2400      Inf     0.9500      NaN     100.00
0.45         0.0000      1.0000      0.2000      Inf     1.0000      NaN     100.00
1.45         0.0000      1.0000      0.2000      Inf     1.0000      NaN     100.00
--------------------------------------------------------------------------------------------------

1) Max Sensitivity Cut-off point= 0.01
2) Max Specificity Cut-off point= 0.33
3) Cost effective Cut-off point= 0.29
4) Max Efficiency Cut-off point= 0.20
>>
```

Minakshy Ramachandran
A20396350
Samruddhi Naik
A20381084

10

F) CONCLUSION AND FINDINGS

→ Hence the data set are loaded as probe and gallery, and their templates were found and hamming distance was calculated for Matching

→ From the above experiment, the test cases (25) showed good ROC and CMC curves

→ It also has good AUC (Area of Under Curve)

AUC = 0.82188.

→ And the standard. area of the error which is 0.09 which is a really good system

→ Confidence interval = 0.64320, which makes it a good test

→ Also the sensitivity, specificity and efficiency, PLR, NLR is calculated by varying the thresholds

NOTE : All of the above data are attached ~~in~~ as screenshot ~~folder~~.