Manotej Meka

CSE 491 – Biometrics

Project 1 Report

1. Genuine and Impostor Scores

|  |  |  |
| --- | --- | --- |
| **Count** | **Genuine** | **Impostor** |
| **Finger Print** | 450 | 450 |
| **Hand** | 450 | 450 |

1. Maximum and Minimum Score

|  |  |  |
| --- | --- | --- |
| **Scores** | **Maximum** | **Minimum** |
| **Finger Print** | 966 | 0 |
| **Hand** | 626 | 0 |

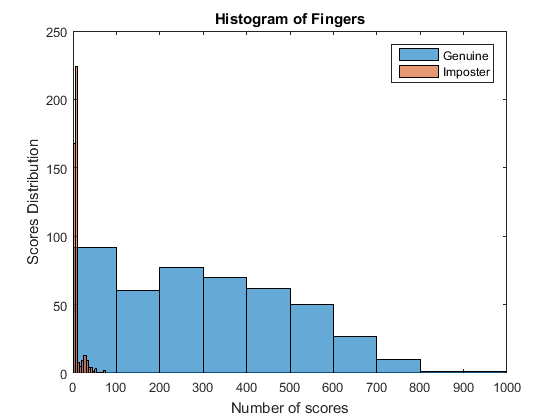
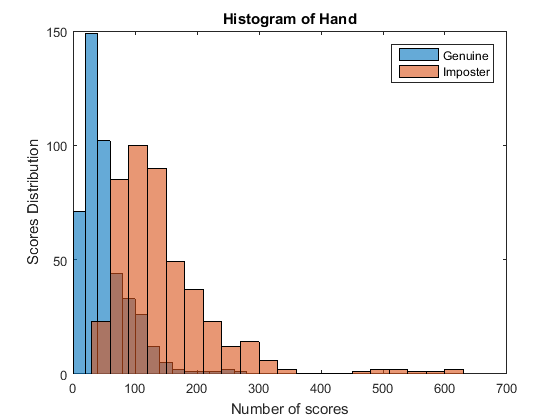
1. Mean and Variance

|  |  |  |
| --- | --- | --- |
| **Finger Print** | **Mean** | **Variance** |
| **Genuine** | 306.58 | 40916 |
| **Imposter** | 7.9711 | 91.008 |

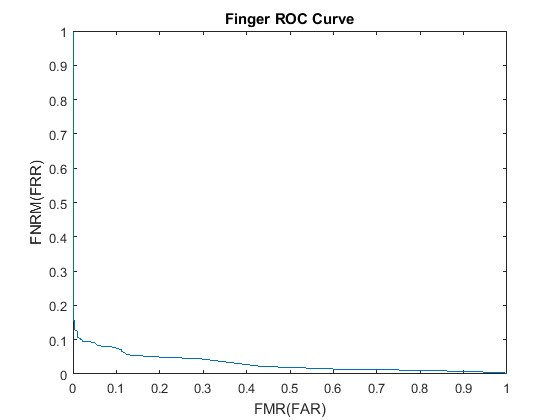
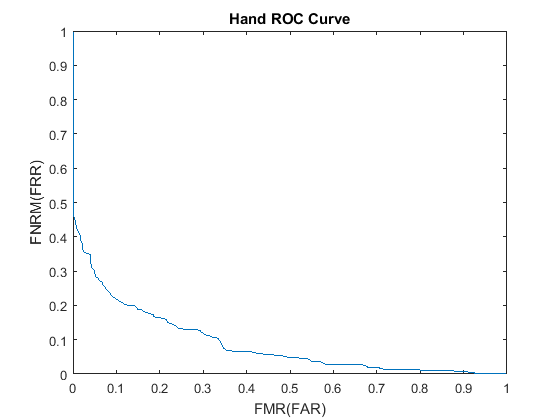
|  |  |  |
| --- | --- | --- |
| **Hand** | **Mean** | **Variance** |
| **Genuine** | 50.644 | 1519.7 |
| **Imposter** | 144.44 | 6941.1 |

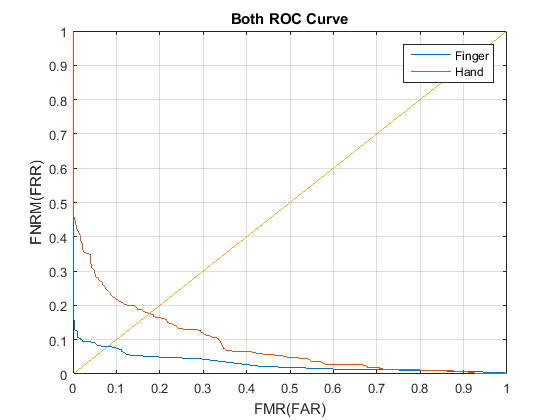
1. D-Prime

|  |  |
| --- | --- |
| **Scores** | **D-Prime** |
| **Finger Print** | 2.0854 |
| **Hand** | 1.442 |

1. Histograms
2. FMR and FNMR

|  |  |  |
| --- | --- | --- |
| Threshold | FMR (\*100 for %) | FNMR (\*100 for %) |
| Fingerprint (32) | 0.04222 (4.22%) | 0.09333 (9.34%) |
| Hand (45) | 0.0044444 (0.45%) | 0.44889 (44.89%) |

1. ROC and EER



|  |  |  |
| --- | --- | --- |
|  | Finger | Hand |
| EER @ 45 degrees | 8 | 17.6 |

1. FNRM @ FMR

|  |  |  |  |
| --- | --- | --- | --- |
| FMNR @ | FMR 10 % | FMR 1% | FMR 0.1% |
| Fingerprint | 7.56 | 12.4 | 15.56 |
| Hand | 22.22 | 42.0 | 46.45 |

1. Opinion Matcher
   1. After looking at all of the above statistics and graphs, the fingerprint matcher performed better than the hand matcher. To start of fingerprint matcher has the higher D’ value meaning a better match. The ROC for the fingerprint is lower through the various thresholds compared the hand. Lastly, the EER matcher is lower for the finger at 8 instead of 17.6 for the hand.

Project 1 Code

clear;

clc;

close all;

% Opening the files

Finger\_Gen\_File = 'finger\_genuine.score';

Finger\_Imp\_File = 'finger\_impostor.score';

Hand\_Gen\_File = 'hand\_genuine.score';

Hand\_Imp\_File = 'hand\_impostor.score';

formatSpec = '%f';

% Reading and extracting files

Gen\_Finger\_Array = fscanf(fopen(Finger\_Gen\_File,'r'),formatSpec);

Imp\_Finger\_Array = fscanf(fopen(Finger\_Imp\_File,'r'),formatSpec);

Gen\_Hand\_Array = fscanf(fopen(Hand\_Gen\_File,'r'),formatSpec);

Imp\_Hand\_Array = fscanf(fopen(Hand\_Imp\_File,'r'),formatSpec);

fclose('all');

Gen\_Fing\_Count = length(Gen\_Finger\_Array);

Imp\_Fing\_Count = length(Imp\_Finger\_Array);

Gen\_Hand\_Count = length(Gen\_Hand\_Array);

Imp\_Hand\_Count = length(Imp\_Hand\_Array);

fprintf('\tTotal number of Genuine and Imposter scores:\n')

disp(table({'Genuine';'Imposter'},[Gen\_Fing\_Count;Imp\_Fing\_Count],[Gen\_Hand\_Count;Imp\_Hand\_Count],'VariableNames',{'TotalCount' 'Finger' 'Hand'}))

%----------------------------------------------------------------------------

% Min and Max

Fin\_max = max(vertcat(Gen\_Finger\_Array,Imp\_Finger\_Array));

Fin\_min = min(vertcat(Gen\_Finger\_Array,Imp\_Finger\_Array));

Han\_max = max(vertcat(Gen\_Hand\_Array,Imp\_Hand\_Array));

Han\_min = min(vertcat(Gen\_Hand\_Array,Imp\_Hand\_Array));

fprintf('\nMax and Min scores generated by each matcher:\n')

disp(table({'Maximum';'Minimum'},[Fin\_max;Fin\_min],[Han\_max;Han\_min],'VariableNames',{'Scores' 'Finger' 'Hand'}))

%----------------------------------------------------------------------------

% Finger Mean and Variance

Fin\_Gen\_mean = mean(Gen\_Finger\_Array);

Fin\_Gen\_vari = var(Gen\_Finger\_Array);

Fin\_Imp\_mean = mean(Imp\_Finger\_Array);

Fin\_Imp\_vari = var(Imp\_Finger\_Array);

fprintf('\nFinger Mean and Variance:\n')

disp(table({'Genuine';'Imposter'},[Fin\_Gen\_mean;Fin\_Imp\_mean],[Fin\_Gen\_vari;Fin\_Imp\_vari],'VariableNames',{'Finger' 'Mean' 'Variance'}))

%----------------------------------------------------------------------------% Hand Mean and Variance

Han\_Gen\_mean = mean(Gen\_Hand\_Array);

Han\_Gen\_vari = var(Gen\_Hand\_Array);

Han\_Imp\_mean = mean(Imp\_Hand\_Array);

Han\_Imp\_vari = var(Imp\_Hand\_Array);

fprintf('\nHand Mean and Variance:\n')

disp(table({'Genuine';'Imposter'},[Han\_Gen\_mean;Han\_Imp\_mean],[Han\_Gen\_vari;Han\_Imp\_vari],'VariableNames',{'Hand' 'Mean' 'Variance'}))

%----------------------------------------------------------------------------

%D Prime Function

Finger\_dprime = DPrime(Fin\_Gen\_mean,Fin\_Imp\_mean,Fin\_Gen\_vari,Fin\_Imp\_vari);

Hand\_dprime = DPrime(Han\_Gen\_mean,Han\_Imp\_mean,Han\_Gen\_vari,Han\_Imp\_vari);

fprintf('\n D-prime Matcher scores for Hand and Finger:\n')

disp(table({'Finger';'Hand'},[Finger\_dprime;Hand\_dprime],'VariableNames',{'Scores' 'DPrime'}))

%----------------------------------------------------------------------------

% Histogram Finger

histogram(Gen\_Finger\_Array);

hold on

histogram(Imp\_Finger\_Array);

legend('Genuine', 'Imposter')

title('Histogram of Fingers')

xlabel('Number of scores')

ylabel('Scores Distribution')

%Histogram Hand

figure;

histogram(Gen\_Hand\_Array);

hold on

histogram(Imp\_Hand\_Array);

legend('Genuine', 'Imposter')

title('Histogram of Hand')

xlabel('Number of scores')

ylabel('Scores Distribution')

%----------------------------------------------------------------------------

% FNMR and FMR

Finger\_FNMR = FNMR(Gen\_Finger\_Array,32);

Finger\_FMR = FMR(Imp\_Finger\_Array,32);

Hand\_FNMR = FNMRDIS(Gen\_Hand\_Array,45);

Hand\_FMR = FMRDIS(Imp\_Hand\_Array,45);

fprintf('\nFinger and Hand FMR,FNMR Calculations:\n')

disp(table({'Finger';'Hand'},[Finger\_FMR;Hand\_FMR],[Finger\_FNMR;Hand\_FNMR],'VariableNames',{'Threshold' 'FMR' 'FNMR'}))

%----------------------------------------------------------------------------

%ROC Curves

fprintf('\nCalclating the ROC curve with the given genuine and impostor scores: \n')

in = input('Would you like to provide Genuine and Imposter scores set [Y/N]: ','s');

if(in == 'Y')

gen = input('Genuine Scores file: ','s');

imp = input('Imposter Scores file: ','s');

option = input('What kind of scores are they: \n1)Distance\n2)Similarity\n Type 1 or 2: ','s')

GEN = fscanf(fopen(gen,'r'),formatSpec);

IMP = fscanf(fopen(imp,'r'),formatSpec);

Start\_Array = min([GEN;IMP]);

End\_Array = max([GEN;IMP]);

if(Start\_Array == 0)

Start\_Array = 1;

end

if(End\_Array == 0)

End\_Arrya = 1;

end

Hand\_Gen\_ROC = 1:End\_Array;

Hand\_Imp\_ROC = 1:End\_Array;

if(option == '1')

[Hand\_Imp\_ROC, Han\_Gen\_ROC] = ROCDIS(GEN,IMP,Start\_Array,End\_Array);

elseif(option == '2')

[Hand\_Imp\_ROC, Han\_Gen\_ROC] = ROC(GEN,IMP,Start\_Array,End\_Array);

end

if(EER\_Number ~= -1)

fprintf('EER Number = %d\n',EER\_Number);

else

fprintf('ERR Number does not exisit!\n');

end

plot(Hand\_Imp\_ROC,Hand\_Gen\_ROC)

title('Input ROC Curve')

xlabel('FMR(FAR)')

ylabel('FNRM(FRR)')

end

in2 = input('Display the ROC Curves of Hand and Finger? [Y/N]: ','s');

if(in2 == 'Y')

fprintf('\nHand and Fingers ROC Curves. \nHand:\n')

Start\_Array = min([Gen\_Hand\_Array;Imp\_Hand\_Array]);

End\_Array = max([Gen\_Hand\_Array;Imp\_Hand\_Array]);

if(Start\_Array == 0)

Start\_Array = 1;

end

if(End\_Array == 0)

End\_Arrya = 1;

end

Hand\_Gen\_ROC = 1:End\_Array;

Hand\_Imp\_ROC = 1:End\_Array;

[Hand\_Imp\_ROC, Hand\_Gen\_ROC] = ROCDIS(Gen\_Hand\_Array,Imp\_Hand\_Array,Start\_Array,End\_Array);

figure;

plot(Hand\_Imp\_ROC,Hand\_Gen\_ROC)

title('Hand ROC Curve')

xlabel('FMR(FAR)')

ylabel('FNRM(FRR)')

fprintf('\nFinger:\n')

Start\_Array = min([Gen\_Finger\_Array;Imp\_Finger\_Array]);

End\_Array = max([Gen\_Finger\_Array;Imp\_Finger\_Array]);

if(Start\_Array == 0)

Start\_Array = 1;

end

if(End\_Array == 0)

End\_Arrya = 1;

end

Fin\_Gen\_ROC = 1:End\_Array;

Fin\_Imp\_ROC = 1:End\_Array;

[Fin\_Imp\_ROC, Fin\_Gen\_ROC] = ROC(Gen\_Finger\_Array,Imp\_Finger\_Array,Start\_Array,End\_Array);

figure;

plot(Fin\_Imp\_ROC,Fin\_Gen\_ROC)

title('Finger ROC Curve')

xlabel('FMR(FAR)')

ylabel('FNRM(FRR)')

figure;

x = 0:1;

y = x;

plot(Fin\_Imp\_ROC,Fin\_Gen\_ROC)

hold on

plot(Hand\_Imp\_ROC,Hand\_Gen\_ROC)

hold on

plot(x,y)

grid on

title('Both ROC Curve')

legend('Finger', 'Hand')

xlabel('FMR(FAR)')

ylabel('FNRM(FRR)')

end

%----------------------------------------------------------------------------

%----------------------------------------------------------------------------

function [ dprime ] = DPrime( mean\_gen, mean\_imp, var\_gen, var\_imp )

%UNTITLED2 Summary of this function goes here

% Detailed explanation goes here

num = sqrt(2)\*abs((mean\_gen-mean\_imp));

den = sqrt((var\_gen)+(var\_imp));

dprime = num/den;

end

%----------------------------------------------------------------------------

%----------------------------------------------------------------------------

function [ fnmr ] = FNMR( gen\_scores ,threshold )

%UNTITLED3 Summary of this function goes here

% Detailed explanation goes here

count = 0;

for i = 1:length(gen\_scores)

if(gen\_scores(i) < threshold)

count = count + 1;

end

end

fnmr = (count/length(gen\_scores));

end

%----------------------------------------------------------------------------

%----------------------------------------------------------------------------

function [ fmr ] = FMR( imp\_scores ,threshold )

%UNTITLED3 Summary of this function goes here

% Detailed explanation goes here

count = 0;

for i = 1:length(imp\_scores)

if(imp\_scores(i) >= threshold)

count = count + 1;

end

end

fmr = (count/length(imp\_scores));

end

%----------------------------------------------------------------------------

%----------------------------------------------------------------------------

function [fmr\_array, fnrm\_array] = ROC( scores\_gen,scores\_imp,start,finish)

%UNTITLED Summary of this function goes here

% Detailed explanation goes here

fnrm\_array = finish;

fmr\_array = finish;

eer\_number = -1;

for i = start:finish

fmr\_array(i) = FMR(scores\_imp,i);

fnrm\_array(i) = FNMR(scores\_gen,i);

end

end

%----------------------------------------------------------------------------

%----------------------------------------------------------------------------

function [ fnmr ] = FNMRDIS( gen\_scores ,threshold )

%UNTITLED3 Summary of this function goes here

% Detailed explanation goes here

count = 0;

for i = 1:length(gen\_scores)

if(gen\_scores(i) >= threshold)

count = count + 1;

end

end

fnmr = (count/length(gen\_scores));

end

%----------------------------------------------------------------------------

%----------------------------------------------------------------------------

function [ fmr ] = FMRDIS( imp\_scores ,threshold )

%UNTITLED3 Summary of this function goes here

% Detailed explanation goes here

count = 0;

for i = 1:length(imp\_scores)

if(imp\_scores(i) < threshold)

count = count + 1;

end

end

fmr = (count/length(imp\_scores));

end

%----------------------------------------------------------------------------

%----------------------------------------------------------------------------

function [fmr\_array, fnrm\_array] = ROCDIS( scores\_gen,scores\_imp,start,finish)

%UNTITLED Summary of this function goes here

% Detailed explanation goes here

fnrm\_array = finish;

fmr\_array = finish;

eer\_number = -1;

for i = start:finish

fmr\_array(i) = FMRDIS(scores\_imp,i);

fnrm\_array(i) = FNMRDIS(scores\_gen,i);

end

end

%----------------------------------------------------------------------------

%----------------------------------------------------------------------------