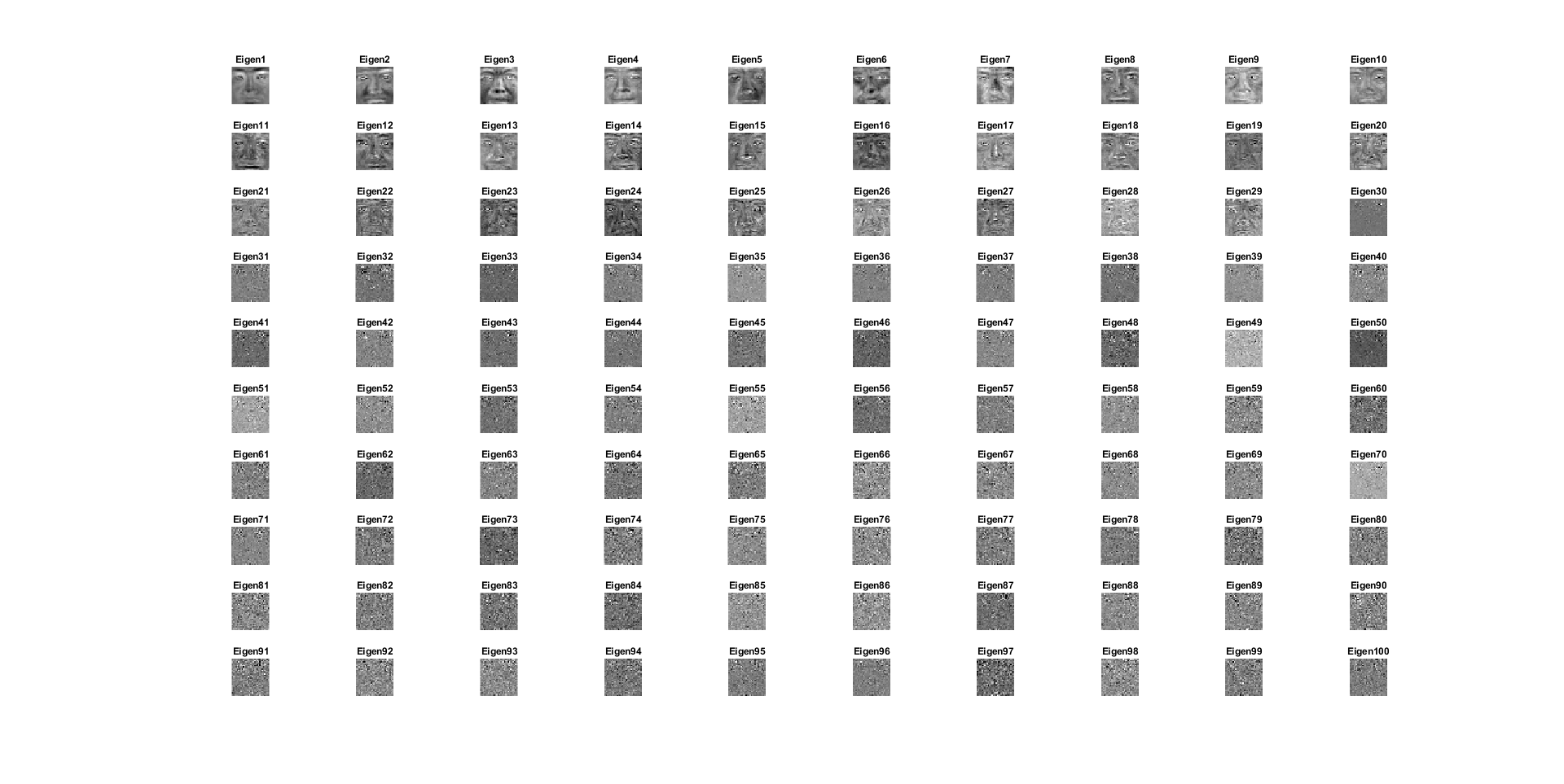
Chen, Kunyu

cse 402

project 3

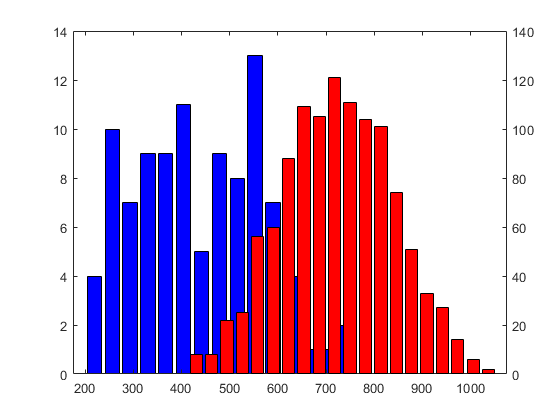
project 3 report

1. Mean Face and the 100 eigen values faces

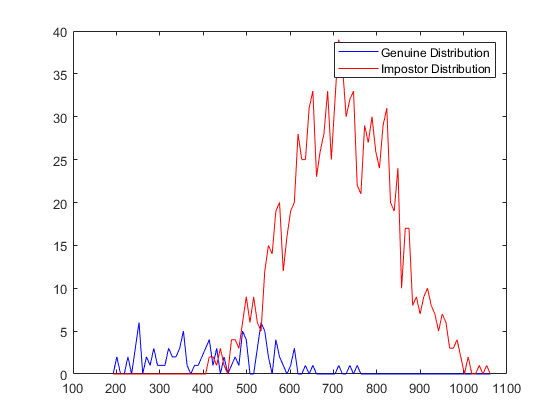
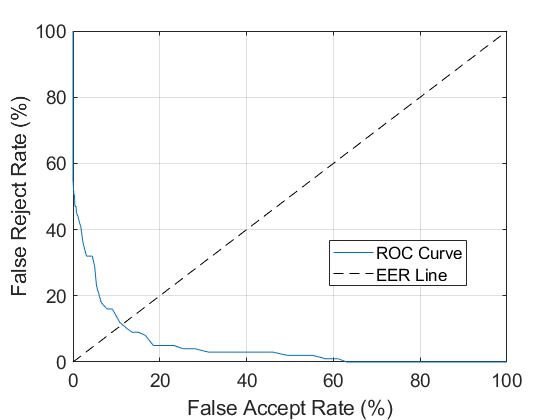




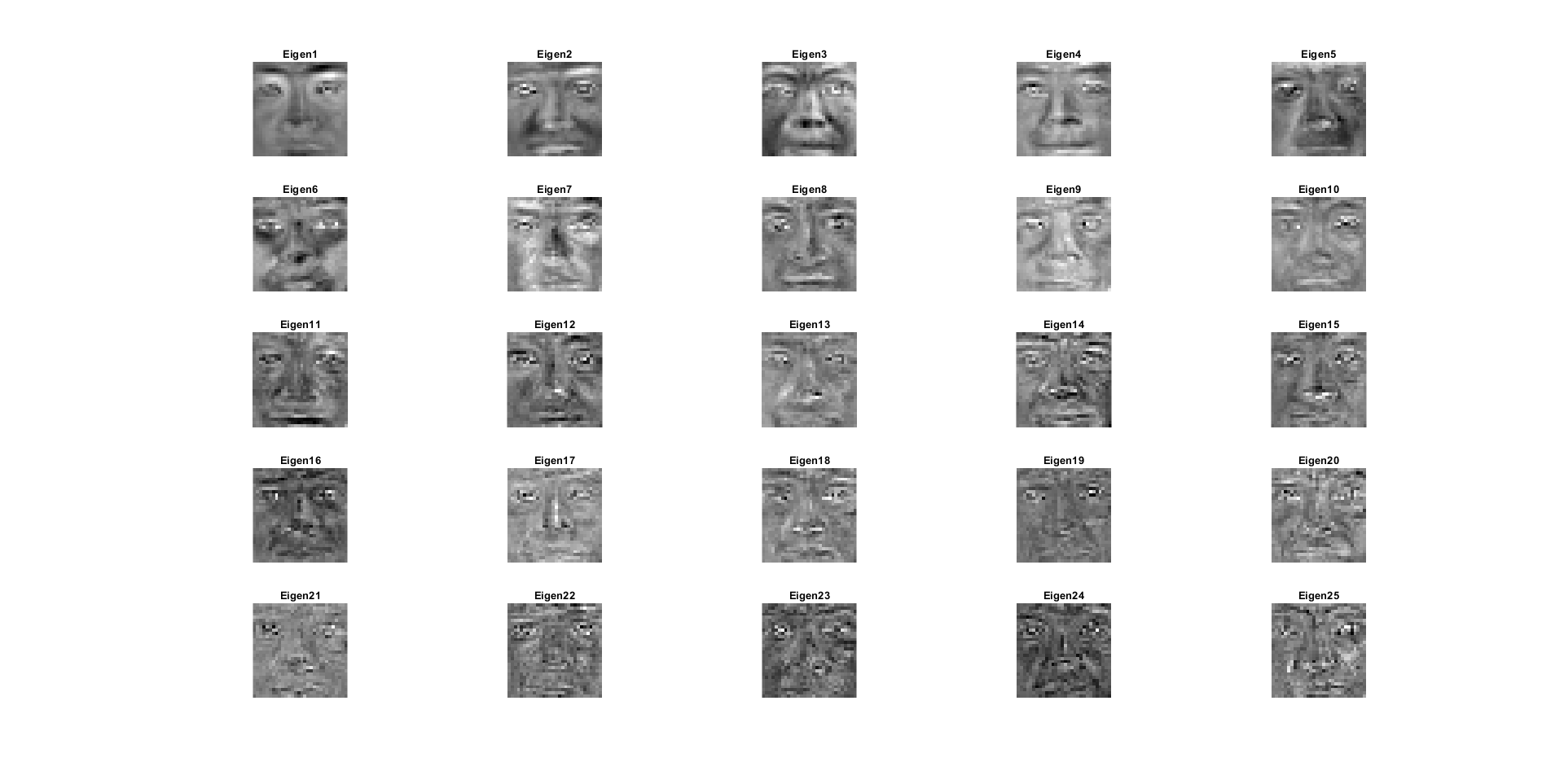
3.Histograms of genuine and impostor scores:



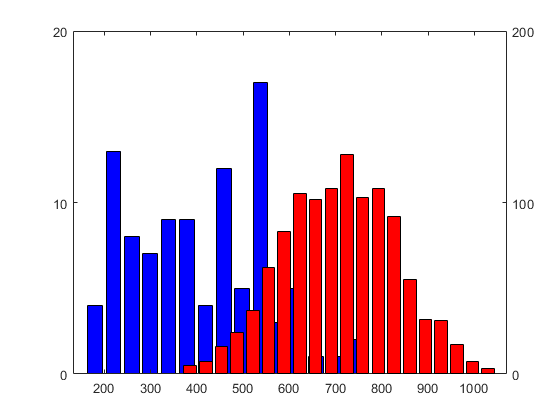
4. ROC, Gen, Imp:

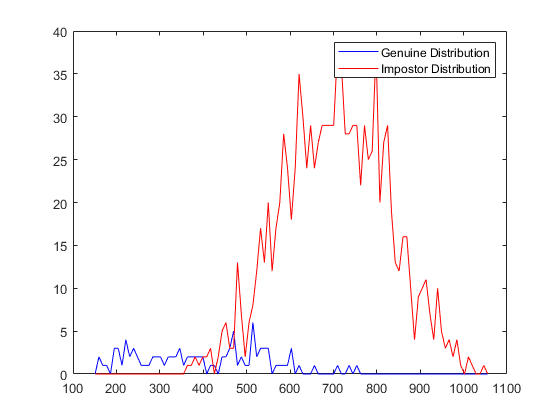
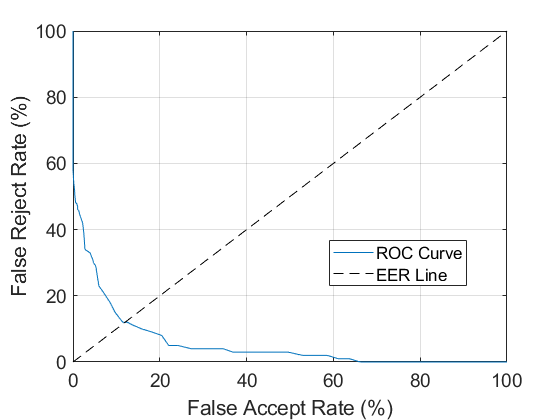
5. Top 25 eigen values faces:



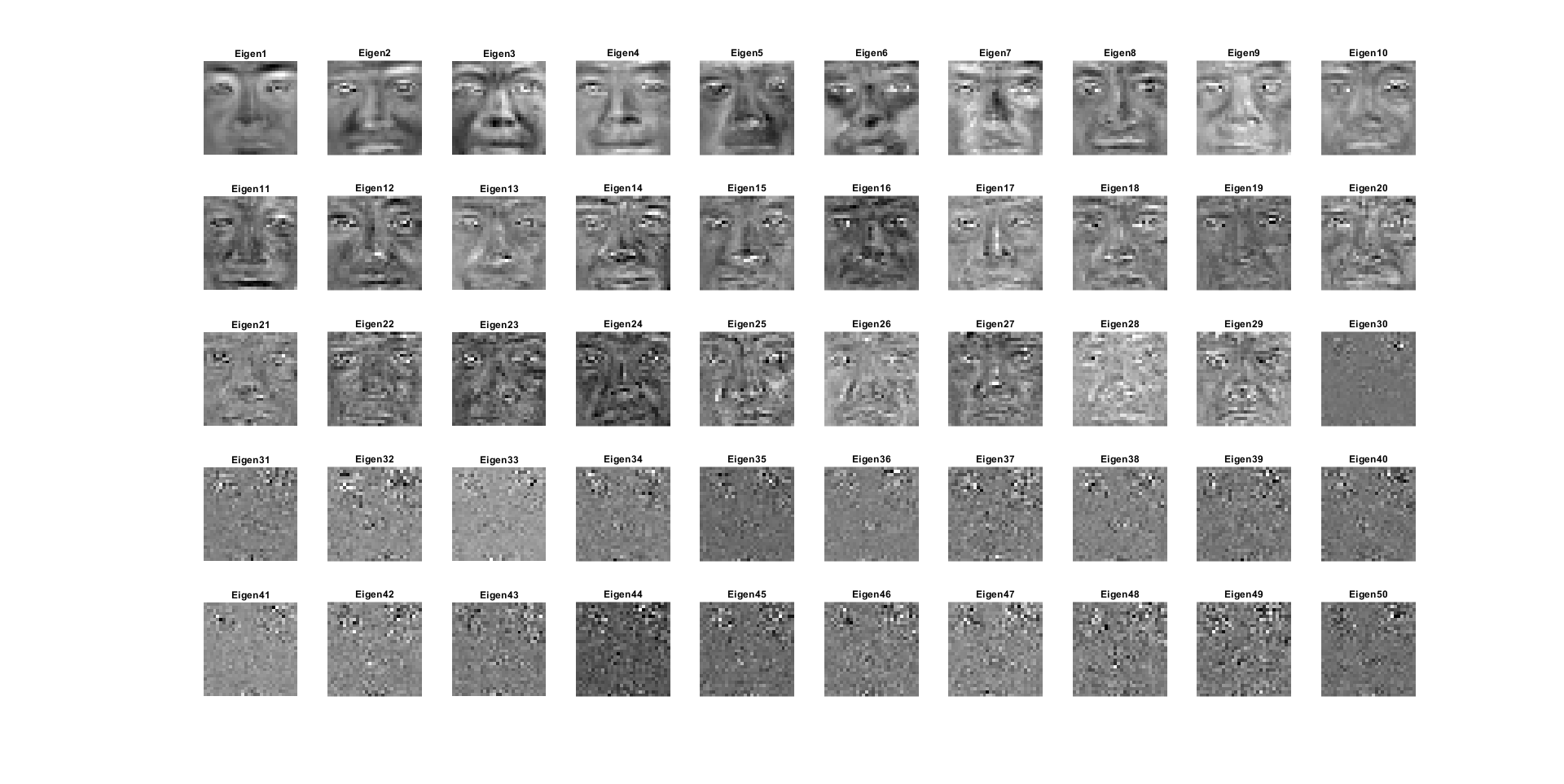
Histograms of genuine and impostor scores:



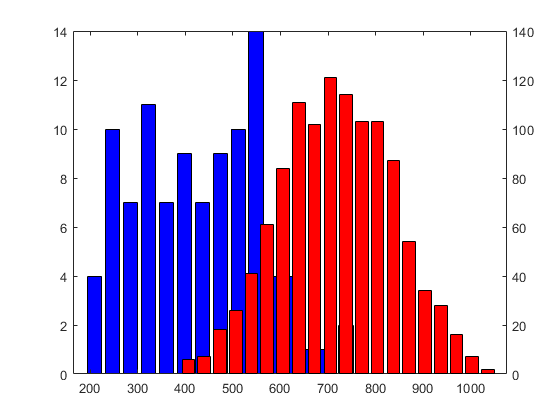
ROC, Gen,Imp:

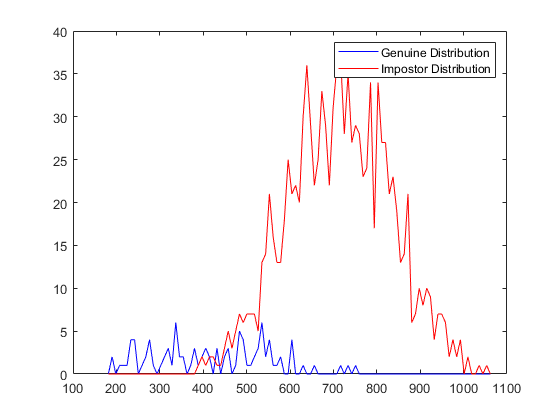
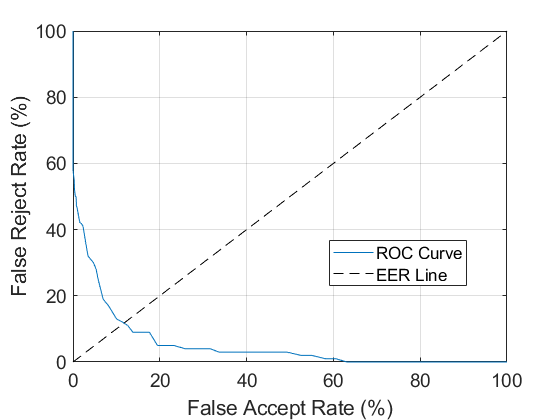
Top 50 eigen values faces:



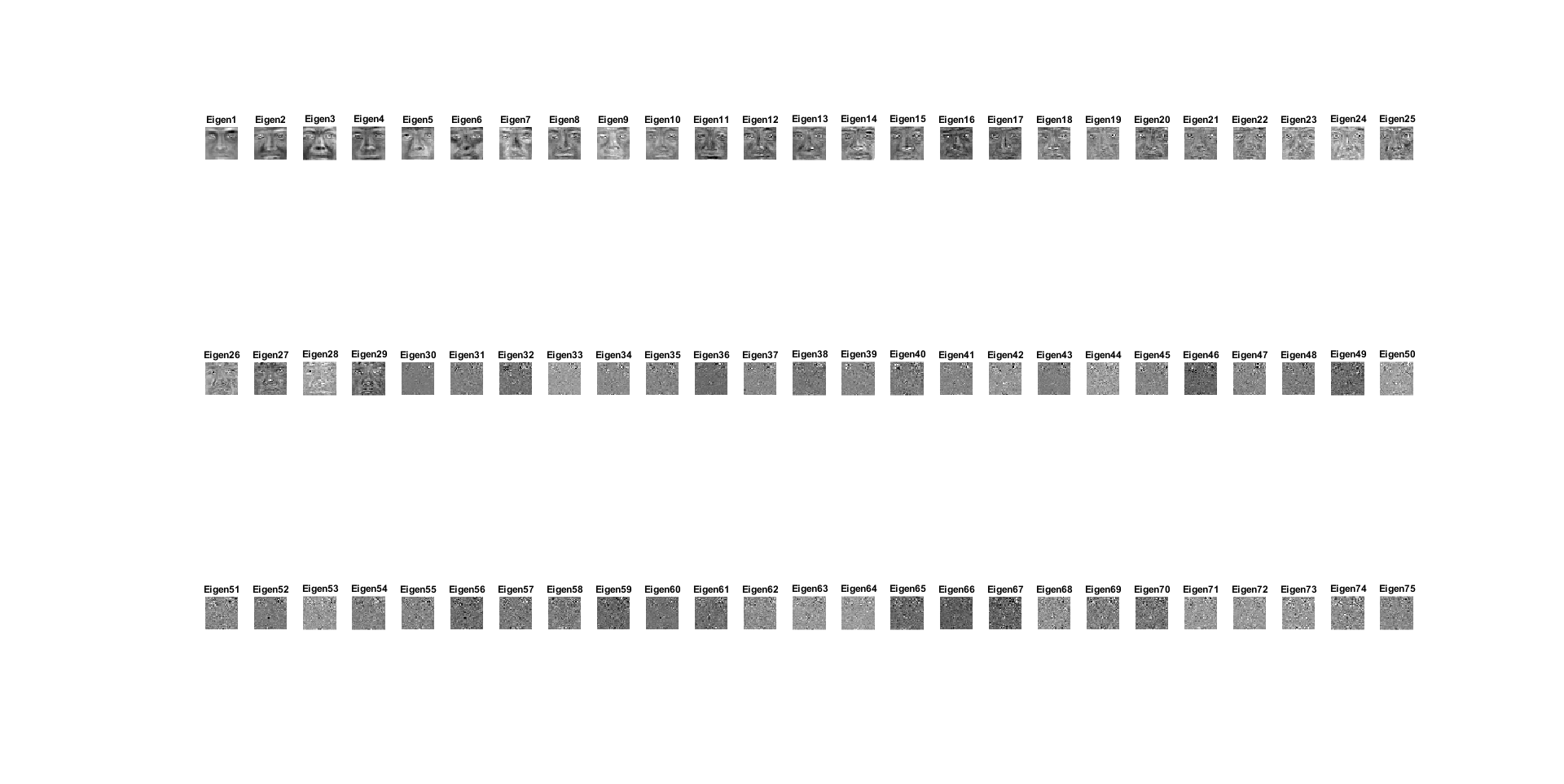
Histograms of genuine and impostor scores:



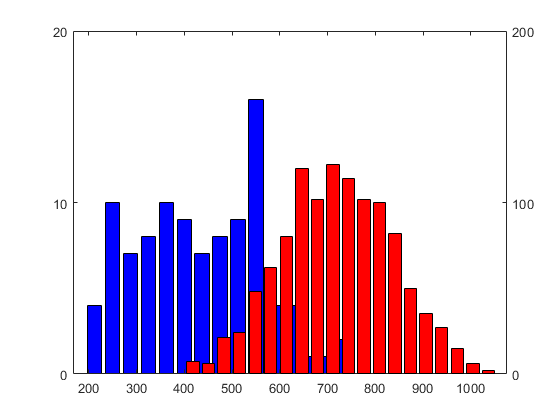
ROC, Gen, Imp:

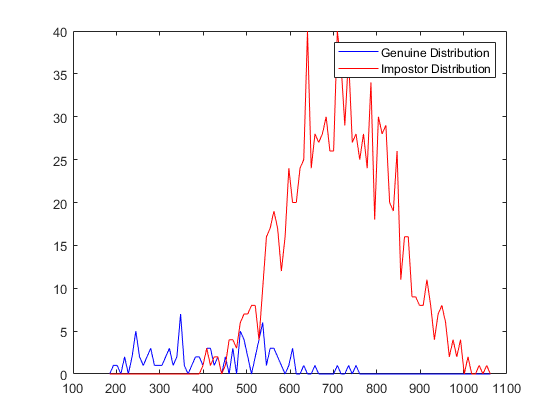
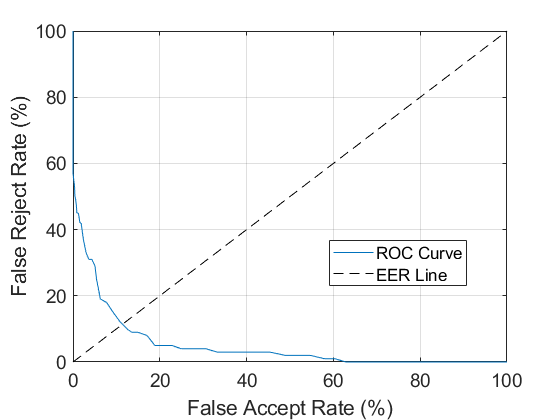
Top 75 eigen values faces:



Histograms of genuine and impostor scores:

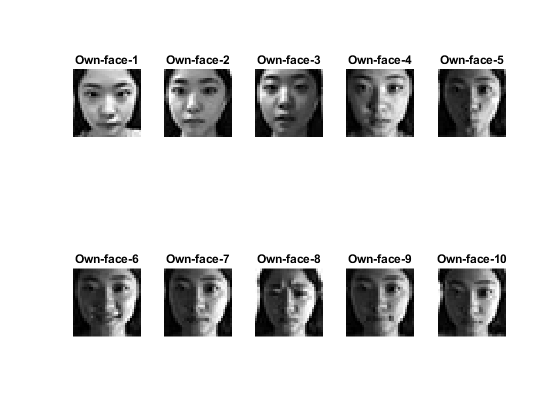


ROC, Gen, Imp:

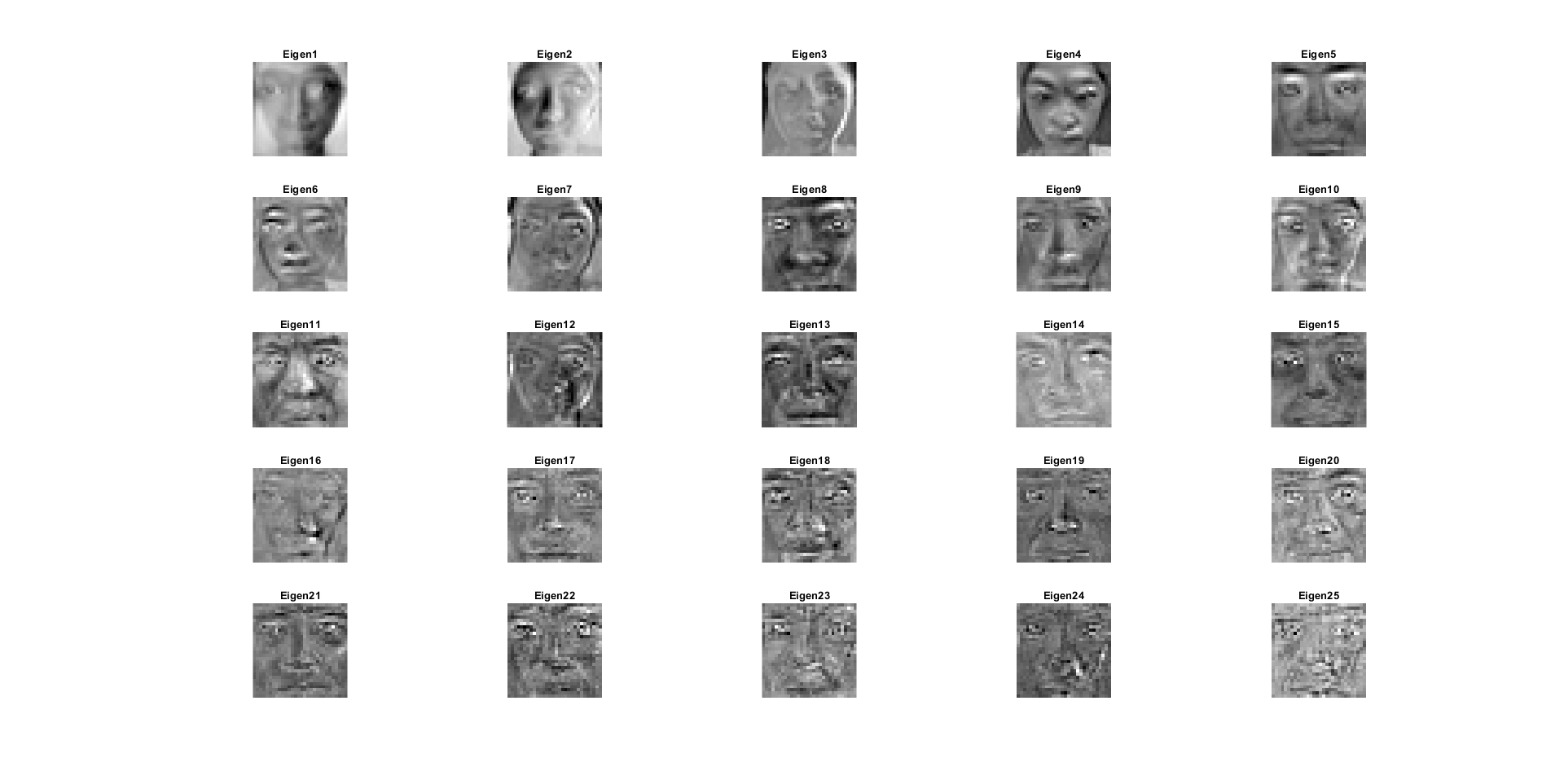
 

According to the ROC curves, we can see that when the number of Top Eigen-face increase, the ROC curve got far from the 0 mark for the FAR and FRR. The FAR and FRR were farther because there are more samples to compare, so there will be more discrepancy and more variation between those images and the mean face.

2. own face:



2.



3.

Columns 1 through 17

2.2243 2.6684 1.9615 2.3045 2.2652 2.2495 2.8720 2.2417 2.2477 1.5466 2.4410 2.5622 2.6593 2.6966 2.9376 2.5984 2.6529

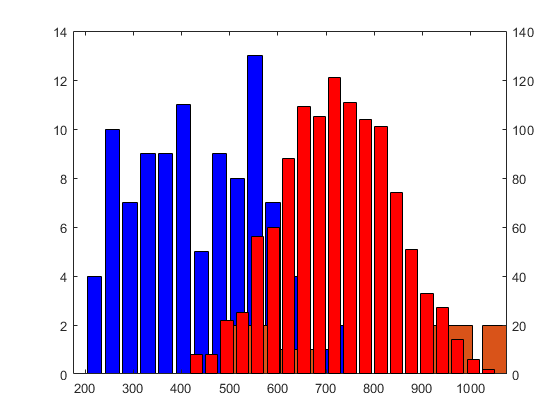
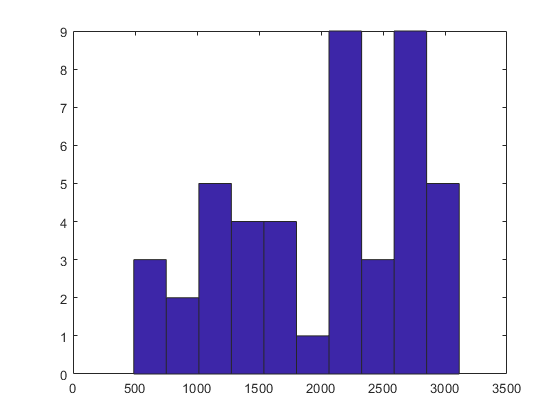
Columns 18 through 34

2.6426 2.7799 2.8074 2.9055 3.1147 2.7467 2.9211 1.3029 1.0844 1.1451 2.2361 1.0550 1.5310 1.2612 1.1254 1.7857 0.9721

Columns 35 through 45

1.0053 0.6807 2.3336 0.5498 1.5499 2.2362 0.4899 1.4475 2.1628 1.6589 1.3568

4. Histograms of genuine:

. 

5. looking at the histograms of my genuine score, I think the face matcher is almost accuracy because the top 15 Eigen-face is cognizable. However, the histograms show that a lot genuine score stay at right, which mean it sees not good enough. I think the reason it is not so good, because the size of those image it 30X30 which may distort the image. The other reason is when people take picture, they do different expression, the long hair and the different light source may cause the machine hard to compare the face.

Code：

clear;

close all;

clc;

%% Pre-Process

files = dir('images/\*.bmp');

count = 1;

%% Getting the face images

for i = 1:length(files)

filename = ['images/' files(i).name];

% 30 faces images

if mod(i,5)<= 3 && mod(i,5) >= 1

img = imread(filename);

images(:,count) = reshape(img,900,1);

count = count + 1;

end

% all faces images

all\_images(:,i) = reshape(imread(filename),900,1);

end

% Showing 30 face images

figure;

for i=1:30

subplot(3,10,i);

imshow(reshape(images(:,i),30,30),[]);

title(['30faces' num2str(i)]);

end

% Each top num eigen-faces

% Process 100 EIGEN-FACE

% [gen,imp] = faceProcess(50,100,10,10, all\_images, images,'T');

% Process 25 EIGEN-FACE

% [gen,imp] = faceProcess(50,25,5,5, all\_images, images,'T');

%

% Process 50 EIGEN-FACE

% [gen,imp] = faceProcess(50,50,5,10, all\_images, images,'T');

% Process 75 EIGEN-FACE

[gen,imp] = faceProcess(50,75,3,25, all\_images, images,'T');

% %% My Images

% Running Own Face Images

files2 = dir('Own\_Face\_Images/\*.jpg');

all\_my\_images = cell(1,10);

count = 1;

figure;

for i = 1:length(files2)

filename = ['Own\_Face\_Images/' files2(i).name];

file2 = imread(filename);

images(:,count) = reshape(img,900,1);

count = count + 1;

subplot(2,5,i);

imshow(file2,[]);

title(['Own-face' num2str(i)]);

all\_my\_images{i} = reshape(file2,900,1);

end

all\_my\_images = cell2mat(all\_my\_images);

[~,~,source] = faceProcess(10,25,5,5, all\_my\_images, images,'F');

figure();

hist(source)

% % [n1,x1] = hist(gen,15);

% % [n2,x2] = hist(imp,20);

% % [ax,h1,h2]=plotyy(x1,n1,x2,n2,@bar,@bar);

% % set(h1,'facecolor','b')

% % set(h2,'facecolor','r')

% % hold on

% % [n3,x3] = hist(source,25);

% % bar(x3,n3)

% % hold off

function [gen,imp,source] = faceProcess( total, num,px, py, all\_images,images,type)

%% a

[E,sumImage ] = calMeanEigenFace(images ,num,px, py );

%% b

% Getting the eigen-coefficients for all the images in the database set

for i = 1:total

coeff(:,i) = E\*(double(all\_images(:,i))-sumImage(:,1));

end

%% c

[ gen,imp, source ] = calGenAndImpScore( coeff, total, type );

%% d

drawROC(gen',imp','d');

end

function [ genuine,impostor, source ] = calGenAndImpScore( coeff, total, type )

%CALGENANDIMPSCORE Calculating genuine scores and impostor scores

% and show histogram

gCount = 1;

iCount = 1;

sCount = 1;

for i= 1:total

for j = i+1:total

temp = sqrt(sum((coeff(:,i)-coeff(:,j)).^2));

% Own face scores

source(sCount) = temp;

sCount = sCount + 1;

if idivide(i-1,int32(5)) == idivide (j-1,int32(5))

% Genuine scores

genuine(gCount) = temp;

gCount = gCount + 1;

else

% Impostor scores

impostor(iCount) = temp;

iCount = iCount + 1;

end

end

end

if type == 'T'

% plot histogram

figure;

[n1,x1] = hist(genuine,15);

[n2,x2] = hist(impostor,20);

[ax,h1,h2]=plotyy(x1,n1,x2,n2,@bar,@bar);

set(h1,'facecolor','b')

set(h2,'facecolor','r')

end

end

function [E,mean\_images ] = calMeanEigenFace(images ,num,px, py )

%GETIMAGES Calculation Mean face calculations and Getting eigen-face

%

%

%% Mean face calculations

for k = 1:900

mean\_images(k,1) = mean(images(k,:));

end

meanFace = reshape(mean\_images,30,30);

figure

imshow(meanFace,[]);

title('Mean Face')

%% Original face subtracted by the mean face

for i = 1: sqrt(length(images))

sub(:,i)=double(images(:,i))-mean\_images(:,1);

end

%% Mean Faces subtracted images

covariance = cov(sub');

%% Getting eigen-face

[V,D] = eigs(covariance,num);

figure;

% Comuting the num eigen faces and displaying them

for i = 1:num

subplot(px,py,i);

imshow(reshape(V(:,i),30,30),[]);

title(['Eigen' num2str(i)]);

end

E = V';

end

function [far, frr] = drawROC(gen,imp,type,varargin)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% [far,frr] = roc(gen,imp, type, varargin)

%% Function to compute ROC (not optimized)

%% gen = all genuine scores (column vector)

%% imp = all impostor scores (column vector)

%% type = 'd' - distance scores; 's' - similarity scores

%% Author: Arun Ross

%% Last Modified: Oct 2006

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('Begin ROC..');

%Determine range of scores

TotGen = length(gen);

TotImp = length(imp);

MinScore = min(min(gen),min(imp));

MaxScore = max(max(gen),max(imp));

Inc = (MaxScore-MinScore)/100;

%Determine histogram of genuine and impostor scores

hgen = histc(gen, MinScore-Inc:Inc:MaxScore+Inc);

himp = histc(imp, MinScore-Inc:Inc:MaxScore+Inc);

figure; plot(MinScore-Inc:Inc:MaxScore+Inc,hgen,'b'); hold on; plot(MinScore-Inc:Inc:MaxScore+Inc,himp,'r');

legend('Genuine Distribution', 'Impostor Distribution');

%The cumulative helps in computing frr/grr at various thresholds

frr = cumsum(hgen);

grr = cumsum(himp);

frr = frr/TotGen\*100;

grr = grr/TotImp\*100;

far = 100 - grr;

gar = 100 - frr;

%Invert definition of far/frr/gar/grr if the scores are distance measures

if (type=='d')

far = 100 - far;

gar = 100 - gar;

frr = 100 - frr;

grr = 100 - grr;

end

%Plot ROC

figure

plot(far, frr, varargin{:})

xlabel('False Accept Rate (%)', 'FontSize',14);

ylabel('False Reject Rate (%)', 'FontSize',14);

set(gca,'FontSize',14);

disp('End ROC..');

%Plot EER Line

hold on; plot(0:100, 0:100, 'k--');

grid on;

legend('ROC Curve', 'EER Line', 'Location', 'Best');