EE 417 Computer Vision

Berkant Deniz Aktaş

19515

Post-Laboratory Report

December 31, 2018

Sabancı University

Faculty of Engineering and Natural Sciences

Computer Science and Engineering

In lab

In this lab, we are asked to develop a face detection and recognition using PCA.

Face Detection

We seperated our data as training and testing set with sizes 360 and 40 respectively. First we processed our dataset to find most significant K eigen faces which has largest eigenvalues. I choose K = 100 because the eigenvalues will converge to 0 after some number. After that I printed most significant 12 eigenfaces.

Figure 1, 12 significant eigenaces with largest eigenvalues

In order to detect a face we need to extract the weighted eigenfaces of an image and add it with mean image aka xbar. Then we will compute our reconstructed image. If the difference between reconstructed image and test image is larger than a threshold it means that detected image is not a face.

Figure 2 Reconstructed image with 5.666 error

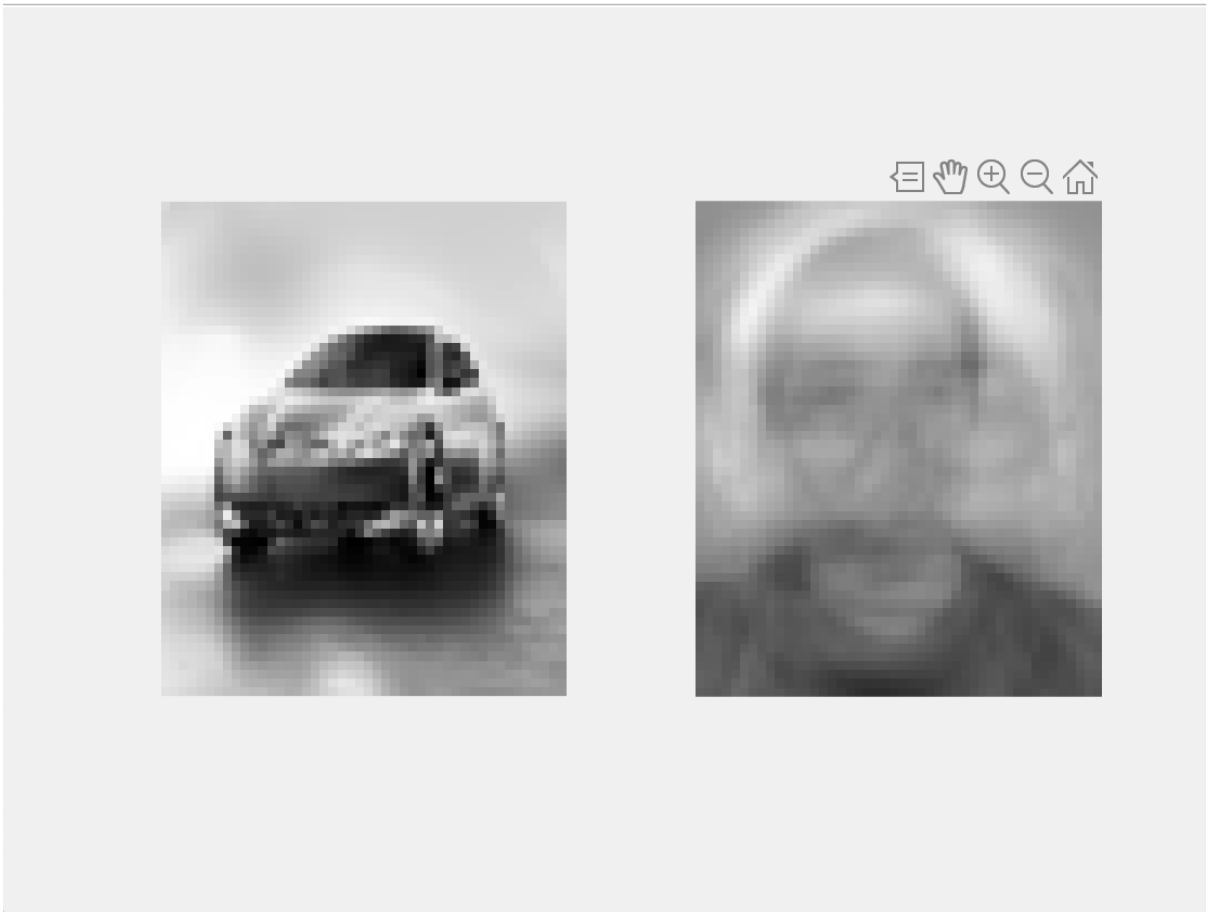

Figure 3 Reconstructed image with 12.9954

With the comparison of Figure 3 and Figure 2 we can see that if the image is not a face, the error will be high and we can conclude that it is not a face.

Face Recognition

In this step, we were asked to develop a database based on the descriptors of images in training set. Then we need to test image and match with a person in the dataset with comparing its descriptors. In this comparision we used eucledian distance and selected minimum euc distance as recognized face.

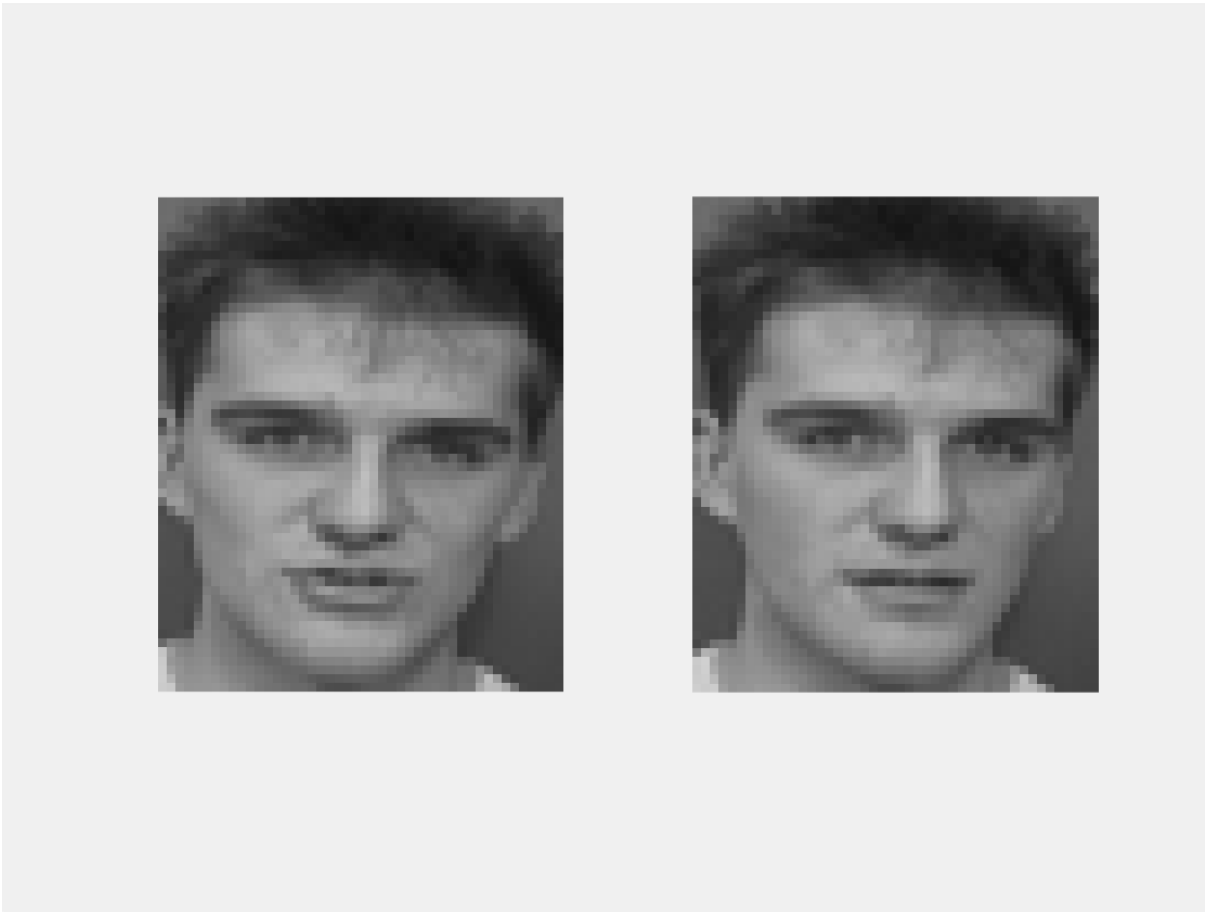

Figure 4. Image 390 is recognized and it is true

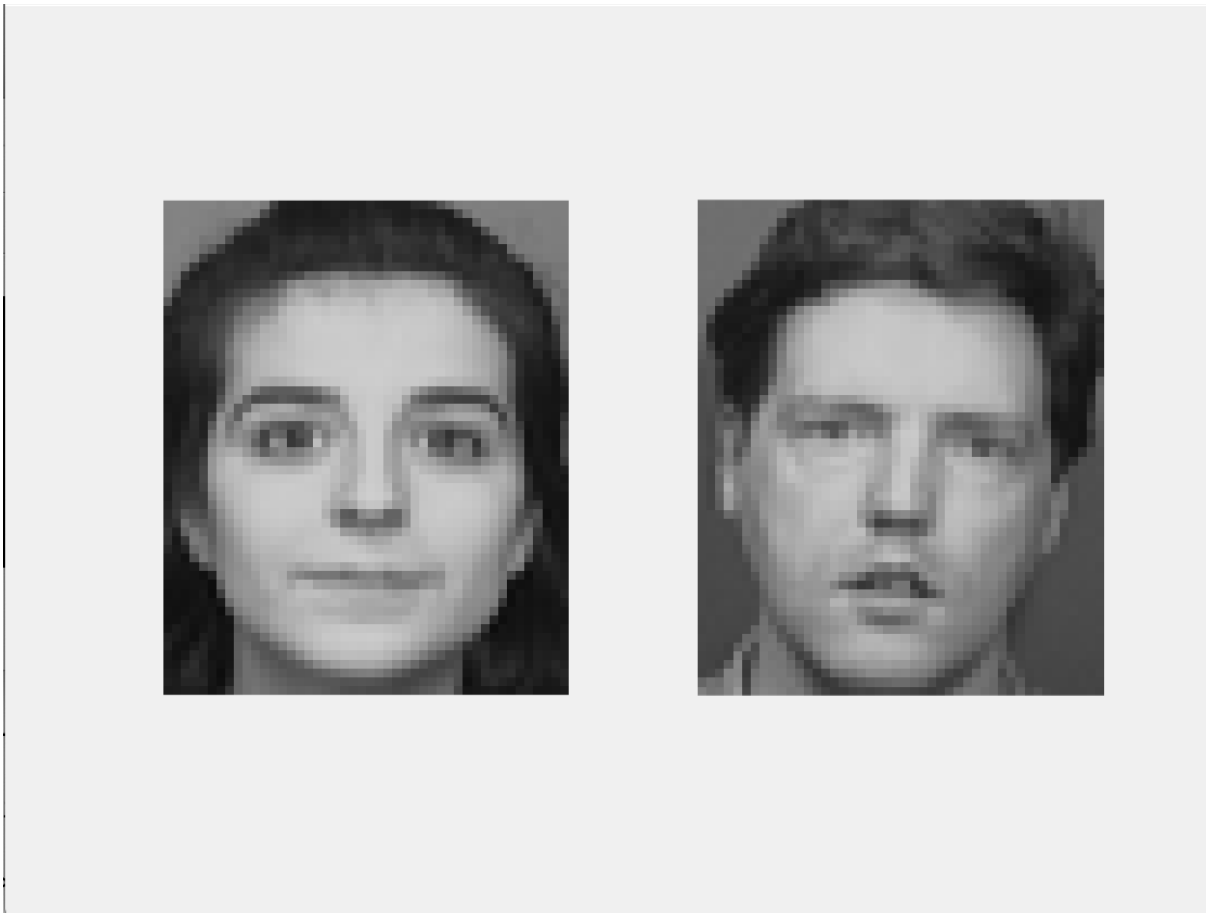

Figure 5. False recognition of image 370

Our classifier detects most of the images but as its shown in the Figure 5, there are some missclassifications. This can be improved with using a higher training set or using different distance formula.

Post lab

For post lab, we are asked to use our own images and use a different distance metric. For this purpose, I took 9 photos of myself and added to training set and added 1 different image to test set. After reshaping images the I obtained following results.

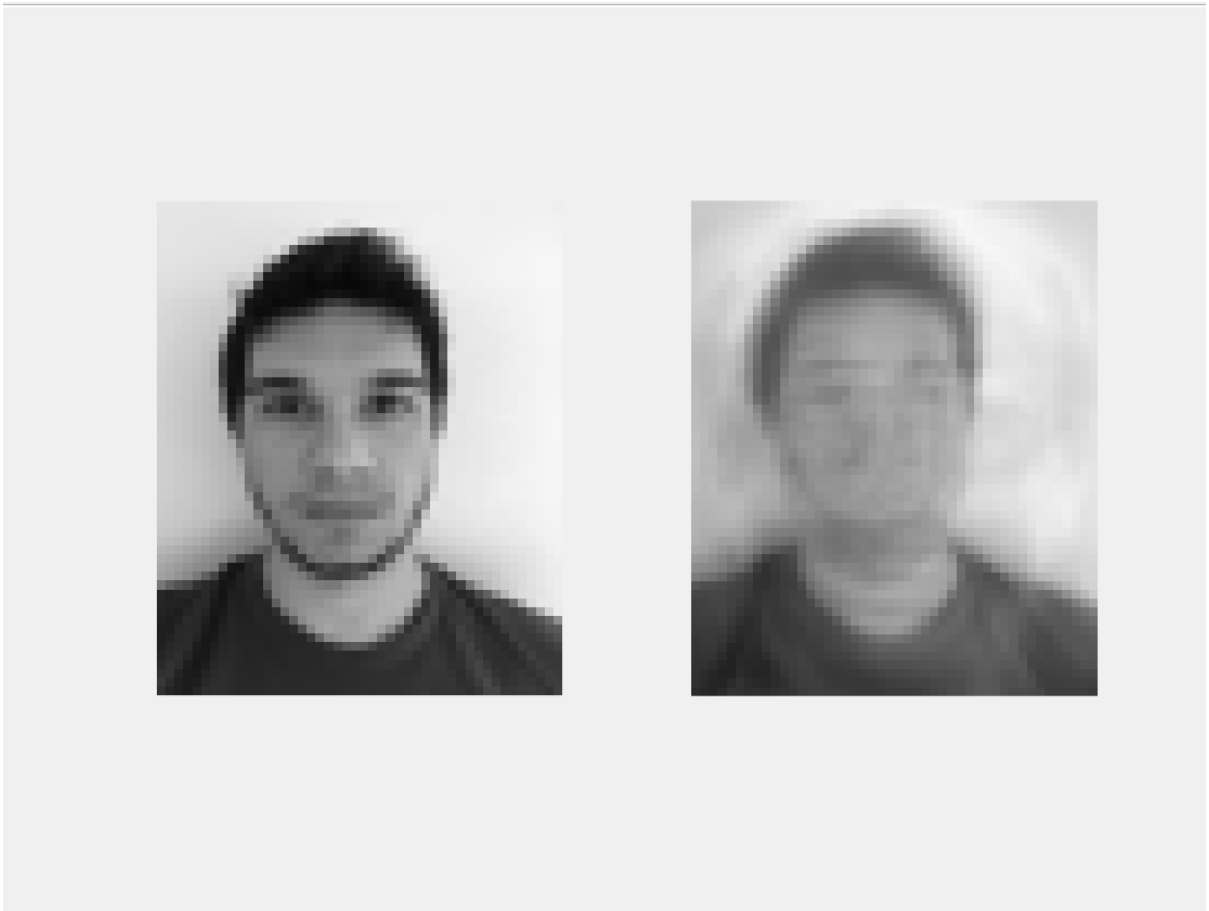

Figure 6. Reconstruction of my face with error 6.86

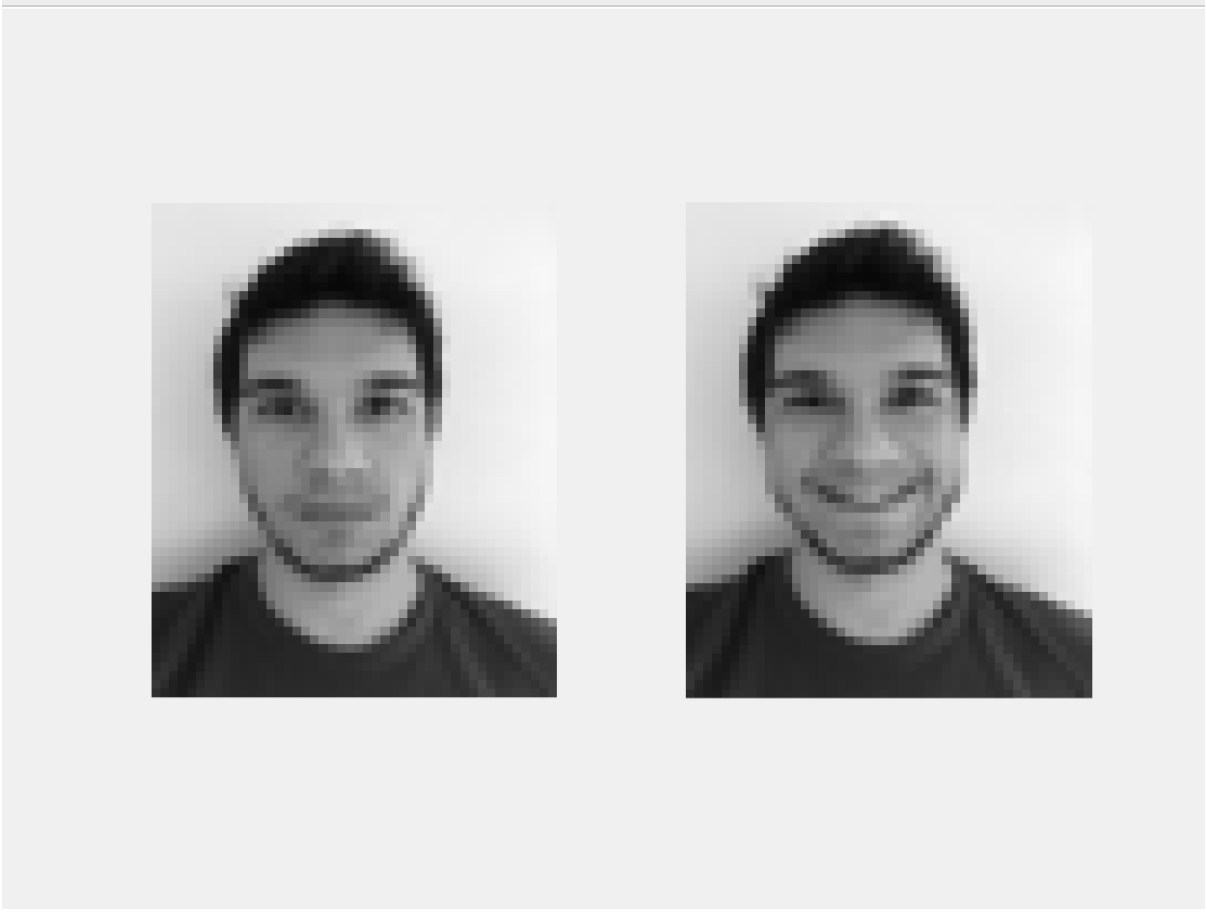

Figure 7 Recognized image of my face

As its shown in the figure 6, It reconstructed my face from eigenfaces with 6.86 error which means it is a face. We can see on Figure 7 that It succesfully recognized my face.

Distance metric

According to research, using Mahalanobis distance gives better results for face recognition, I changed the distance metric to mahalanobis distance and tested the images in test set.

Figure 8. 361st image in dataset recognized by both

Figure 9. both of them recognized wrong image 370

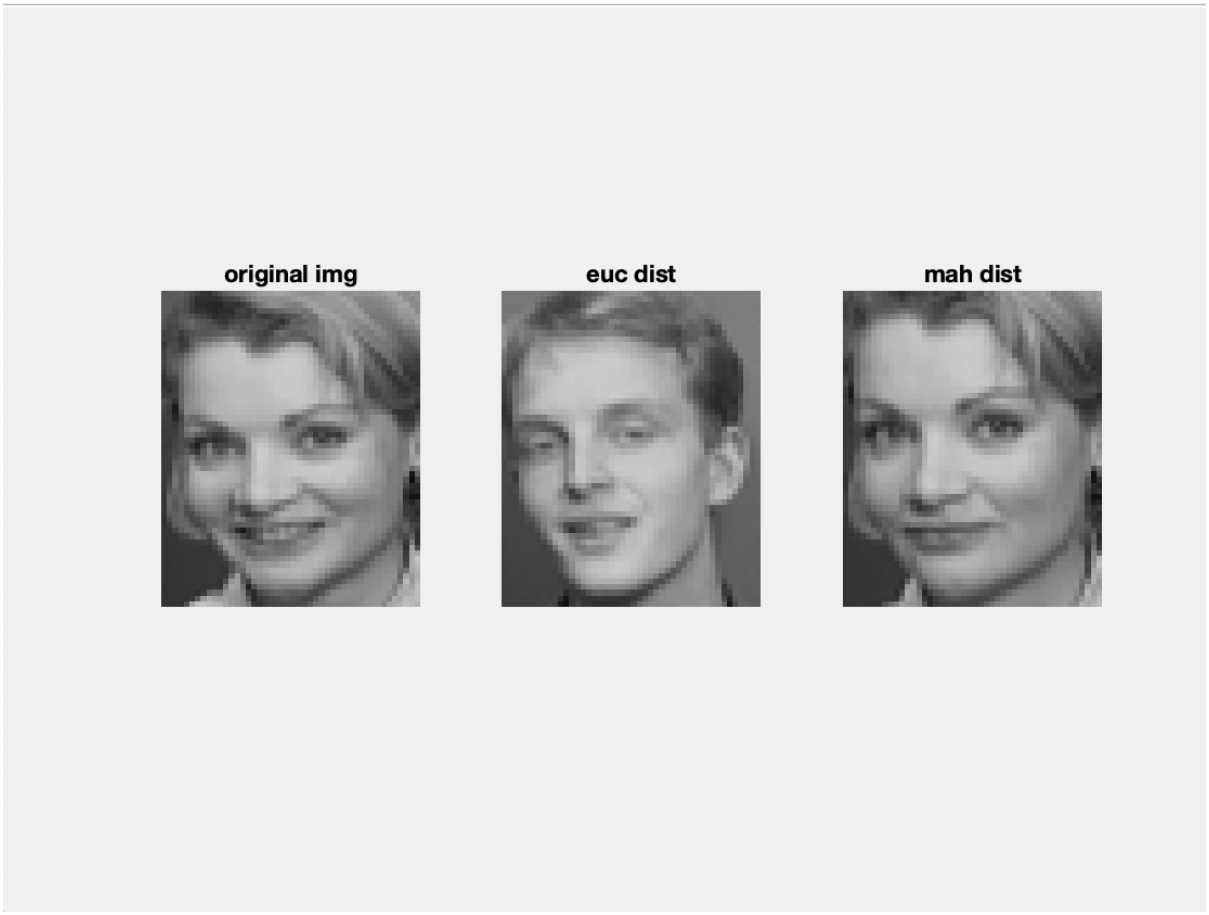

Figure 10. Mah gave true, euc false on image 395th image

I manually tested 40 images in test set. Eucledian distance and mahalanobis distance gave almost same results except Figure 395th image. Mahalanobis distance predicted true face where euc distance predicted wrong image. Both of them failed to detect Image number 370. In a sample size of 40, misclassification rate of eucledian distance is approximately 5% and mahalanobis distance has misclassification rate of 2.5%. However, using a larger dataset for training will decrease errors.

Code

```
clear all; close all; clc;
load('faces.mat');
Seq = faces;
[row,col,num]=size(Seq);
%Image=uint8(reshape(Seq, [row,col]));

berkant1 = imread('img1.jpeg');
berkant1 = imresize(rgb2gray(berkant1),[56 46]);
berkant1 = reshape(berkant1, [], 1);
berkant2 = imread('img2.jpeg');
berkant2 = imresize(rgb2gray(berkant2),[56 46]);
berkant2 = reshape(berkant2, [], 1);
berkant3 = imread('img3.jpeg');
berkant3 = imresize(rgb2gray(berkant3),[56 46]);
berkant3 = reshape(berkant3, [], 1);
berkant4 = imread('img4.jpeg');
berkant4 = imresize(rgb2gray(berkant4),[56 46]);
berkant4 = reshape(berkant4, [], 1);
berkant5 = imread('img5.jpeg');
berkant5 = imresize(rgb2gray(berkant5),[56 46]);
berkant5 = reshape(berkant5, [], 1);
berkant6 = imread('img6.jpeg');
berkant6 = imresize(rgb2gray(berkant6),[56 46]);
berkant6 = reshape(berkant6, [], 1);
berkant7 = imread('img7.jpeg');
berkant7 = imresize(rgb2gray(berkant7),[56 46]);
berkant7 = reshape(berkant7, [], 1);
berkant8 = imread('img8.jpeg');
berkant8 = imresize(rgb2gray(berkant8),[56 46]);
berkant8 = reshape(berkant8, [], 1);
berkant9 = imread('img9.jpeg');
berkant9 = imresize(rgb2gray(berkant9),[56 46]);
berkant9 = reshape(berkant9, [], 1);
berkanttest = imread('imgtest.jpeg');
berkanttest = imresize(rgb2gray(berkanttest),[56 46]);
berkanttest = reshape(berkanttest, [], 1);

berkantmatrix = [berkant1 berkant2 berkant3 berkant4 berkant5 berkant6
berkant7 berkant8 berkant9];

cartest = imread('BMW-TA.jpg');
cartest = imresize(rgb2gray(cartest),[56 46]);
cartest = reshape(cartest, [], 1);


train = Seq(:,1:360);
train = [train berkantmatrix];
train = double(train) / 255;
test = Seq(:,361:400);
test = [test berkanttest cartest];
test = double(test) / 255;


xbar = mean(train');
xbar = xbar';
```

```
A = train-xbar;
K = 100;
[V,D] = eigs(A'*A,K);

newV = (A*V)/norm(A*V);
%35 mah > eig  10 both wrong
exampleimg = test(1:2576,35);


a = [];
minia = exampleimg-xbar;
%minia = minia;
for i = 1:1:K
b = dot(minia,newV(:,i));
a = [a b];
end
reconstruct = xbar + (a*newV')';

first12faces = [];

for i = 1:1:12
eigen1 = a(:,i) * newV(:,i);
eigen1 = eigen1 * 255;

first12faces = [first12faces eigen1];
end


eigfacesimgs = [];
for i=1:1:12
eigfacesimgs = [eigfacesimgs double(  reshape(first12faces(:,i), [56,46])
)];
end
figure; montage(eigfacesimgs, 'Size', [1 1]);

hold on;

% % Printing out 12 eigenfaces
% eigfacesimgs = [];
% for i=1:1:12
% eigfacesimgs = [eigfacesimgs uint8(reshape(eigfaces(:,i)*255,
[row,col]))];
% end
% figure; montage(eigfacesimgs, 'Size', [1 1]);
% %hold on;


error = norm(exampleimg - reconstruct);
disp(error);

exampleimg = reshape(exampleimg,56,46);
reconstruct = reshape(reconstruct,56,46);
figure;
subplot(1,2,1)
```

```
imshow(exampleimg);
subplot(1,2,2)
imshow(reconstruct);
if(error>7.000)
    disp('not face');
else
    disp('face');
end
%%%%%% difference of exampleimg and reconstruct will give reconstruction
%%%%%% error
%
%training = train(1:2576,1:360);
%eigenfaceVec = training'*newV;

eigenfaceVec = [];
for i=1:1:369
    x = train(:,i); % Test face
    diffx = x - xbar;
    dbrow = [];
    for j=1:1:K
        smalla = dot(diffx, newV(:,j));
        dbrow = [dbrow smalla];
    end
    eigenfaceVec = [eigenfaceVec; dbrow];
end


%eucled
[row,col] = size(eigenfaceVec);
matrix= [];
for i = 1:1:369
    sum = 0;
    for j = 1:1:K
        sum = sum + (eigenfaceVec(i,j) - a(j))^2;
    end
    sum = sqrt(sum);
    matrix = [matrix;sum];
end

%mahala
[row,col] = size(eigenfaceVec);
matrix2= [];
for i = 1:1:369
    sum = 0;
    for j = 1:1:K

            sum = sum + ((eigenfaceVec(i,j) - a(j)))^2  / D(j,j);
    end
   % sum = sqrt(sum);
    matrix2 = [matrix2;sum];
end



%minval = min(matrix);
```

```
ind = find(matrix(:) == min(matrix(:)));
ind2 = find(matrix2(:) == min(matrix2(:)));
foundimg = train(1:2576,ind);
foundimg = reshape(foundimg,56,46);
foundimg2 = train(1:2576,ind2);
foundimg2 = reshape(foundimg2,56,46);

figure;
subplot(1,3,1)

imshow(exampleimg);
title('original img');

subplot(1,3,2)

imshow(foundimg);
title('euc dist');

subplot(1,3,3)
imshow(foundimg2);
title('mah dist');
```