EE417

POST-LAB #10 REPORT

Face Reconstruction and Face Recognition

In this lab we will implement a system that reconstruct and recognize faces by use of eigenfaces. But first, let me give some background information to understand the procedure better.

As a starting point, we will treat an N x M image as a vector in NM-dimensional space (form vector by collapsing rows from top to bottom into one long vector). Then we convert x into v1 and v2 coordinates.

$$\mathbf{x} \rightarrow ((\mathbf{x}-\overline{x})\cdot\mathbf{v_1},(\mathbf{x}-\overline{x})\cdot\mathbf{v_2})$$

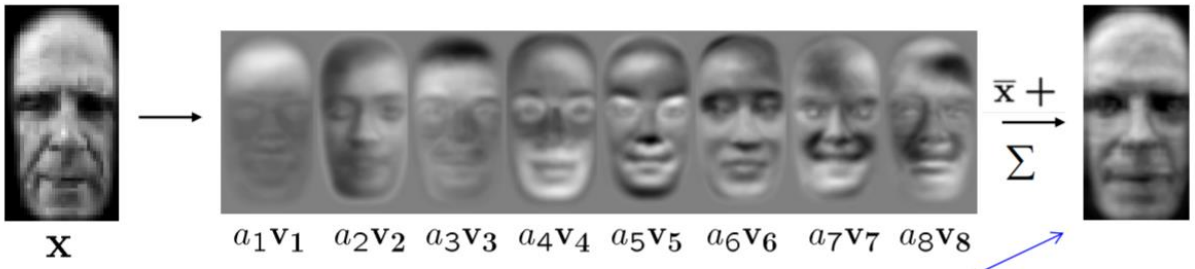
Where v1 measures distance for classification and v2 measures positon to specify the point.

Face Reconstruction

We will use PCA method to achieve face reconstruction. This method introduces dimensionality reduction. The space of all faces is a “subspace” of the space of all images. We suppose that it has K dimension. This K is chosen by the most effective K largest eigenvalues. Large eigenvalue shows the most variation in the face which represents the essential characteristics of the face. Then we compute small a coefficients as weights

$$\mathbf{x} \rightarrow (\underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_1}}_{a_1},\ \underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_2}}_{a_2},\ldots,\ \underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_K}}_{a_K})$$

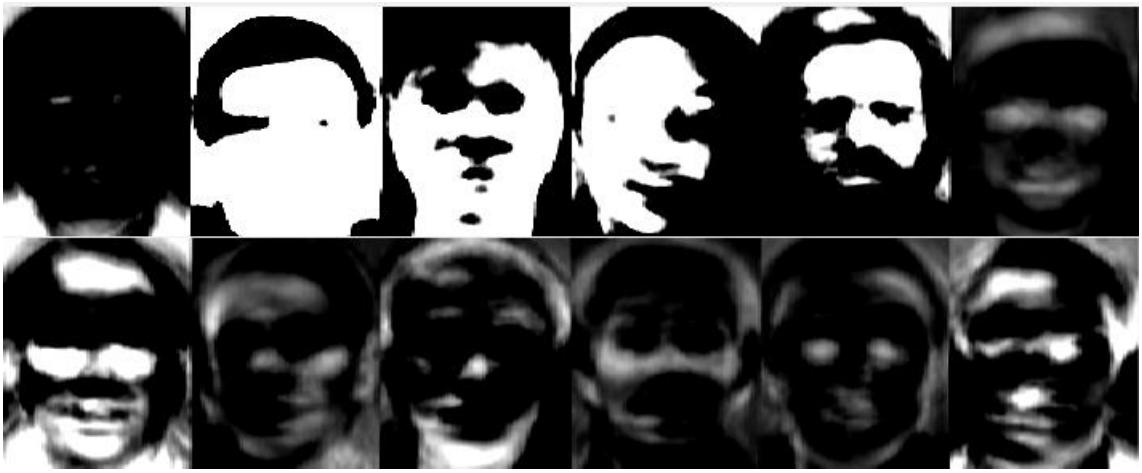
(Compressed representation of face,
K usually much smaller than NM)



Reconstructed face $\mathbf{x} \approx \overline{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_K\mathbf{v_K}$

(a1,a2…) then we multiply them with the faces. Then after, we add xbar which is the mean of the all faces in the training set. We can interpret that the reconstructed face is the weighted sum of the all faces plus mean characteristics of the face that seems really reasonable. In addition to these, we also determined a threshold value by subtracting mean of the faces from the constructed image. Norm of this error should be smaller than our threshold value in order to be detected as a face. We chose threshold as 8 by experimentally.

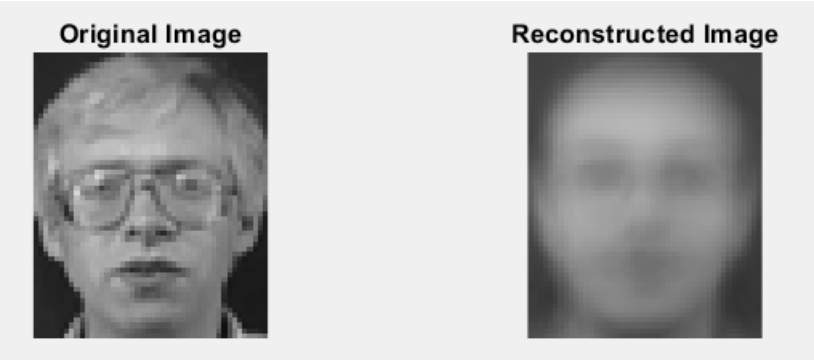
Eigen Faces



The first 12 eigen faces are shown above.

Reconstructed Faces

Face 362



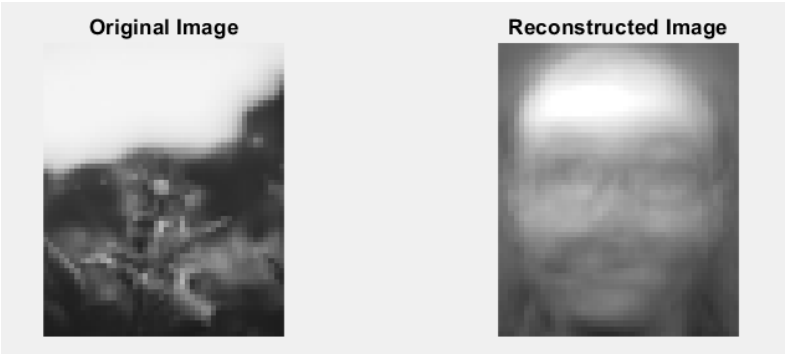
Error is 7.3242. It is a face

Face 382

Error is 6.1560. It is a face.

Face 380

Flower Image (Downloaded from internet)



Error is 10.2538. It is not a face

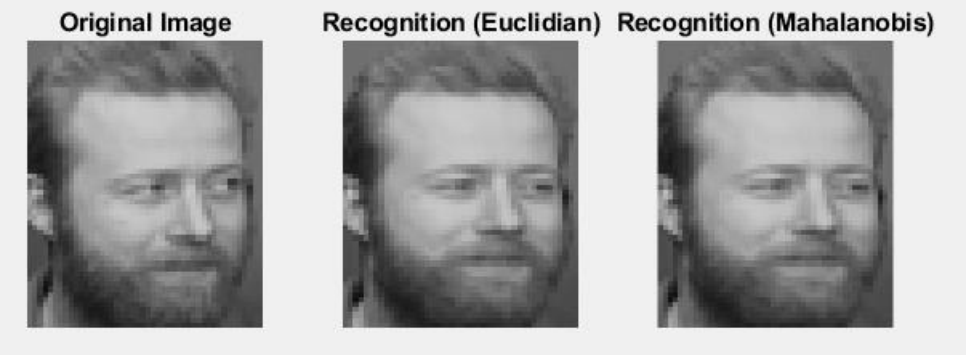
Face Recognition

To achieve face recognition, we will create a database that consists of the descriptors of all the images in the training set (a_1,a_2…a_K). Then we will compute the descriptor of the test image and find the Euclidian distance of it from all the values in the database. Then we will take the image with the smallest Euclidian distance. We can understand that if it has smaller distance, we have more similar features to that image. We also used different distance metric which is called Mahalanobis distance that uses also eigenvalues and sometimes gives better results in comparison with Euclidian distance.

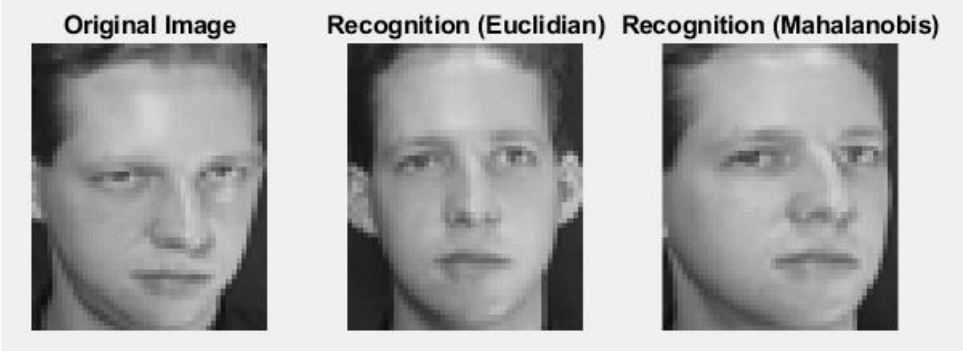
Recognized Faces

Face 380 - CORRECT

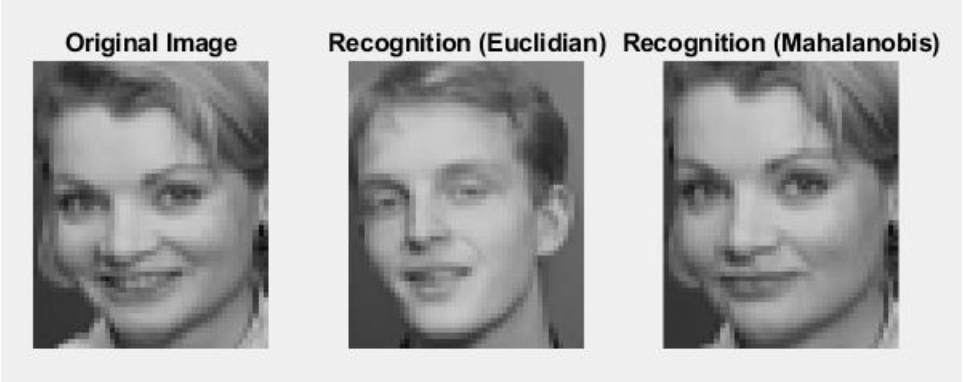
Face 386 - CORRECT



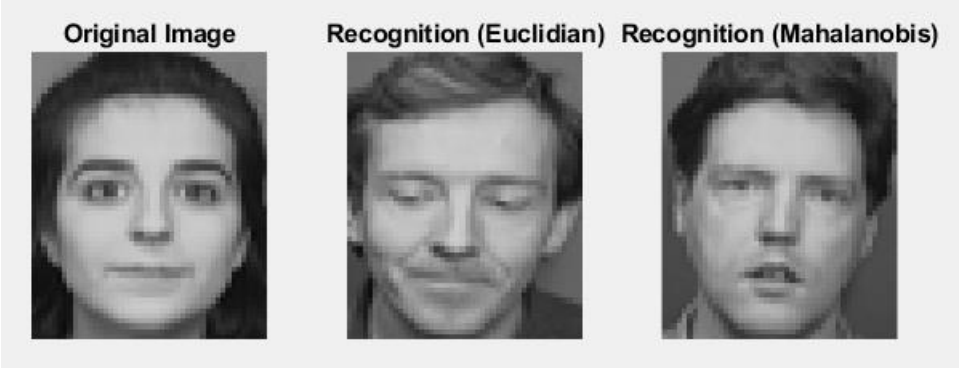
Face 361 - CORRECT



Face 390 – EUC INCORRECT – MAHAL CORRECT



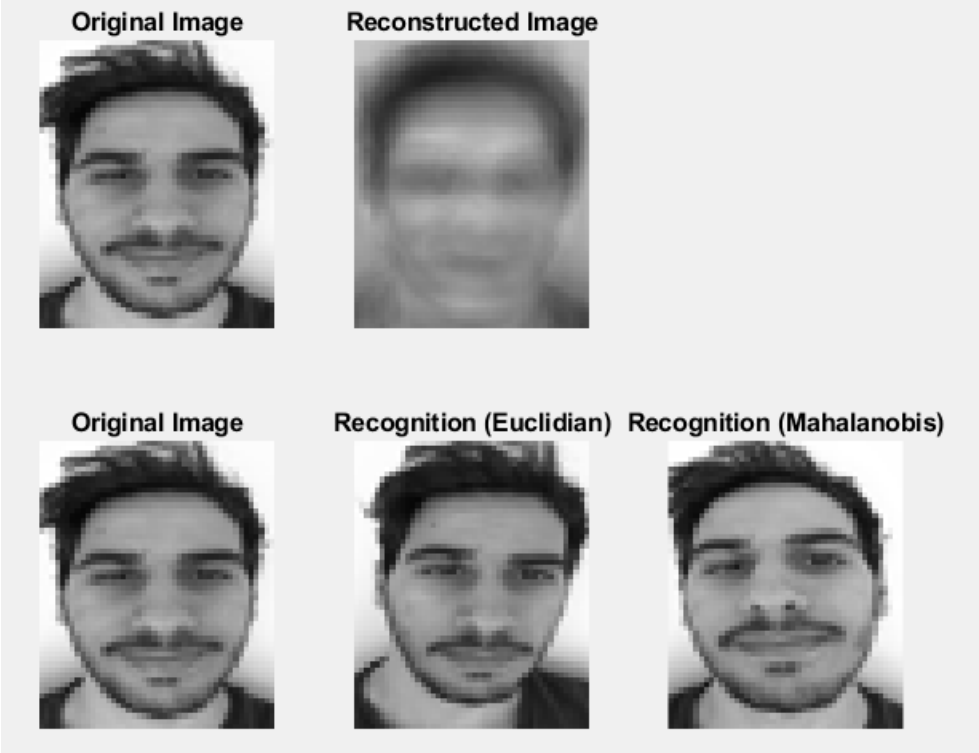
Face 370 – BOTH INCORRECT



After examining of the images, we can say that at faces 380, 386 both distance metrics give the same correct results. In faces 361 Mahalanobis metrics give the correct but different result than Euclidian distance metric. Moreover, in face 390, Euclidian distance give an incorrect result. On the other hand, Mahalanobis gives the correct result. Lastly, both metrics fail to give the correct result in face 370. According to these results, we can say that Mahalanobis gives better results.

Face Reconstruction & Recognition – Our Image

In order to recognize my face, I took 9 photos of myself from different angles and resized them to add to the training set. After training, I took another photo to test. At the end, the system could reconstruct and recognize my face correctly. (I also think that reconstruction has a poor quality.) We also got different correct results by Euclidian and Mahalanobis distance metrics which is quite interesting result for me.



CODE - *lab10.m*

```
clear all; clc;


load('faces.mat');
faces = double(faces)/255;
train = faces(:,1:360);
row = 56, col = 46;

% Loading my own image
alatimgs = [];
for i=1:1:9
    alatimg = imresize(rgb2gray(imread(sprintf('aimg%01d.jpeg',i))),[row
col]);
    alatimg = reshape(alatimg, [], 1);
    alatimgs = [alatimgs alatimg];
end

testalat = imresize(rgb2gray(imread('aimg101.jpeg')),[row col]);
testalat = double(reshape(testalat, [], 1))/255;
alatimgs = double(alatimgs)/255;
```

```
flower = imresize(rgb2gray(imread('flower.jpg')),[row col]);
flower = double(reshape(flower, [], 1))/255;

train = [train alatimgs];
trainsize = 369;
xbar = mean(train')';
A = train - xbar;

%xrec = testalat; % My test face
%xrec = flower; % Test face for not a face
xrec = faces(:,362); % Test face from faces dataset
testx = xrec; % Test Face for Recognition

K = 100;
[V D] = eigs(A'*A,K); % Returns K largest eigenvalues
newV = (A*V)/(norm(A*V));
diffx = xrec - xbar;
faceX = 0;
database = [];

normav = norm(A*V);
% Face Reconstruction
eigfaces = [];
for i=1:1:K
    smalla = dot(diffx, newV(:,i));
    av = (smalla * newV(:,i));
    faceX = faceX + av;
    if i<=12
        eigfaces = [eigfaces av];
    end
end

faceX = faceX + xbar;



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

% Bulding the database for K eigfaces
database = [];
for i=1:1:trainsize
    x = train(:,i); % Test face
    diffx = x - xbar;
    dbrow = [];
    for j=1:1:K
        smalla = dot(diffx, newV(:,j));
        dbrow = [dbrow smalla];
    end
    database = [database; dbrow];
end
```

```
% Calculating of the descriptor
descriptor = [];
for i=1:1:K
    testdiffx = testx - xbar;
    descriptor = [descriptor dot(testdiffx, newV(:,i))];
end

% Face Recognition
eucdistances = [];
for i=1:1:trainsize
    eucdist = 0;
    for j=1:1:K
        eucdist = eucdist + (database(i,j)-descriptor(j))^2;
    end
    eucdistances = [eucdistances; sqrt(eucdist)];
end

% Face Recognition
mahdistances = [];
for i=1:1:trainsize
    mahdist = 0;
    for j=1:1:K
        mahdist = mahdist + ((database(i,j)-descriptor(j))^2)/D(j,j);
    end
    mahdistances = [mahdistances; mahdist];
end

recind = find(eucdistances(:) == min(eucdistances(:)));
mahrecind = find(mahdistances(:) == min(mahdistances(:)));

err = x - faceX;
N = vecnorm(err);
threshold = 8;

    disp(N);
if(N < threshold)
    disp('It is a face');
else
    disp('It is not a face');
end

subplot(2,3,1);
imshow(uint8(reshape(xrec*255, [row,col])));
title('Original Image');
subplot(2,3,2);
imshow(uint8(reshape(faceX*255, [row,col])));
title('Reconstructed Image');

subplot(2,3,4);
imshow(uint8(reshape(testx*255, [row,col])));
title('Original Image');
subplot(2,3,5);
imshow(uint8(reshape(train(:,recind)*255, [row,col])));
title('Recognition (Euclidian)');
subplot(2,3,6);
imshow(uint8(reshape(train(:,mahrecind)*255, [row,col])));
title('Recognition (Mahalanobis)');
```