Machine Learning for Signal Processing

Computer Homework-2

Osman Kaan Kurtça – 040160090

Aziz Gündoğdu – 040160112

Q1 – NASA DECOMPOSİTİON

Codes:

```
clc; clear all; close all;

nasacolor=imread('TarantulaNebula.jpg');
figure; image(nasacolor);

nasa=sum(nasacolor,3,'double');
m=max(max(nasa));
nasa=nasa*255/m; %rgb to gray scale

figure; image(nasa); colormap(gray(256));
title('Grayscale NASA photo');

[U,S,V]=svd(nasa); %we apply svd function to get
eigenfaces(eigenvectors) and eigen values
figure; semilogy(diag(S)); %we plot eigenvalues as
semilogarithmic to observe rapidly dropping off

nasa100=U(:,1:100)*S(1:100,1:100)*V(:,1:100)';
nasa50=U(:,1:50)*S(1:50,1:50)*V(:,1:50)';
nasa25=U(:,1:25)*S(1:25,1:25)*V(:,1:25)';
% we get nasa images that include 25, 50 and 100
eigenfaces.

figure; image(nasa25); colormap(gray(256));
title('25 eigenfaces');

figure; image(nasa50); colormap(gray(256));
title('50 eigenfaces');

figure; image(nasa100); colormap(gray(256));
title('100 eigenfaces');
```

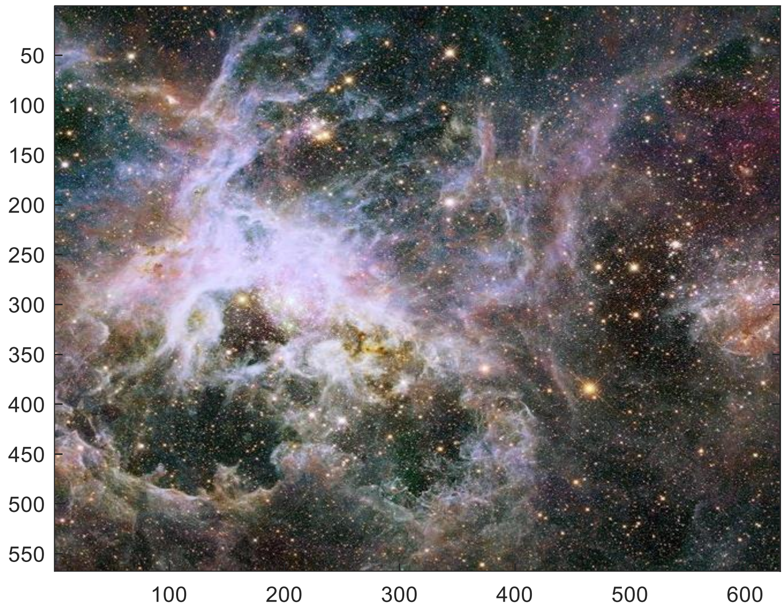


Figure 1- Nasa Original Photo

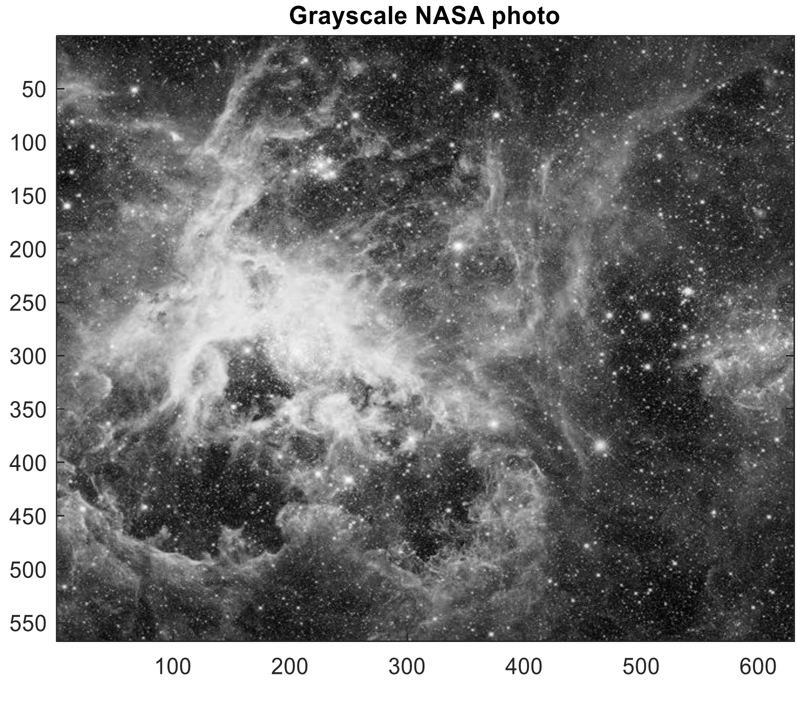


Figure 2- Grayscale Nasa Photo

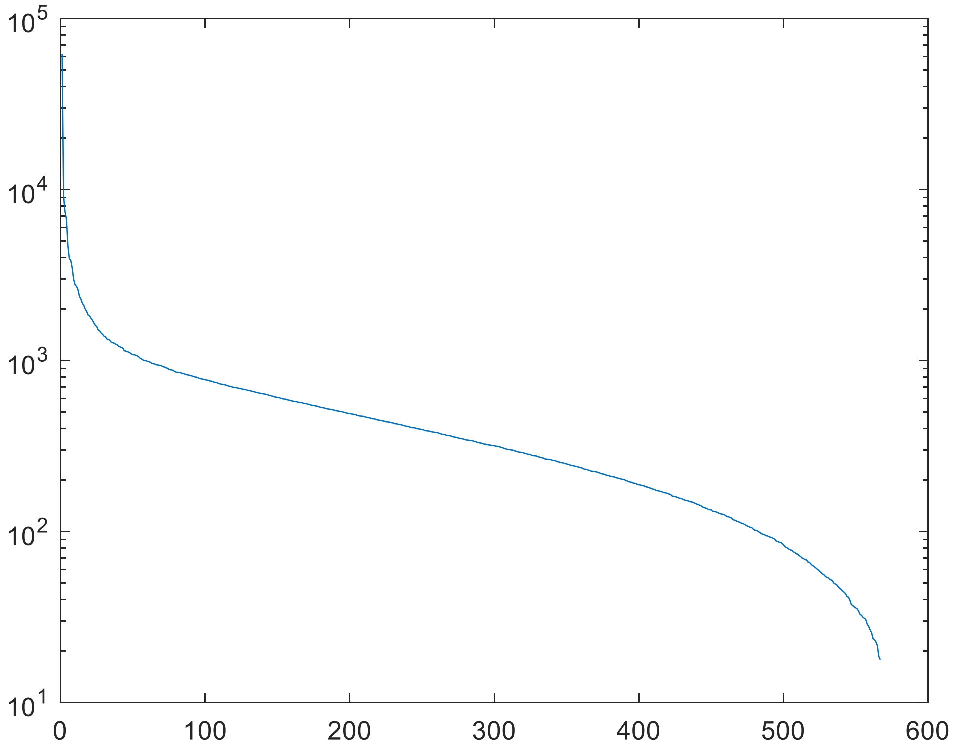


Figure 3- Semilogy Plot of Eigenvalues

We observe that the values drop off very rapidly to less than 2% of the maximum in fewer than 50 values.

On the next page, we see the Nasa images that include 25, 50 and 100 eigenfaces.
We observe only very slight differences between the original and the one with 100 singular values, some noticeable differences with 50 singular values while you should see serious degradation of the image in the case of 25 singular values.

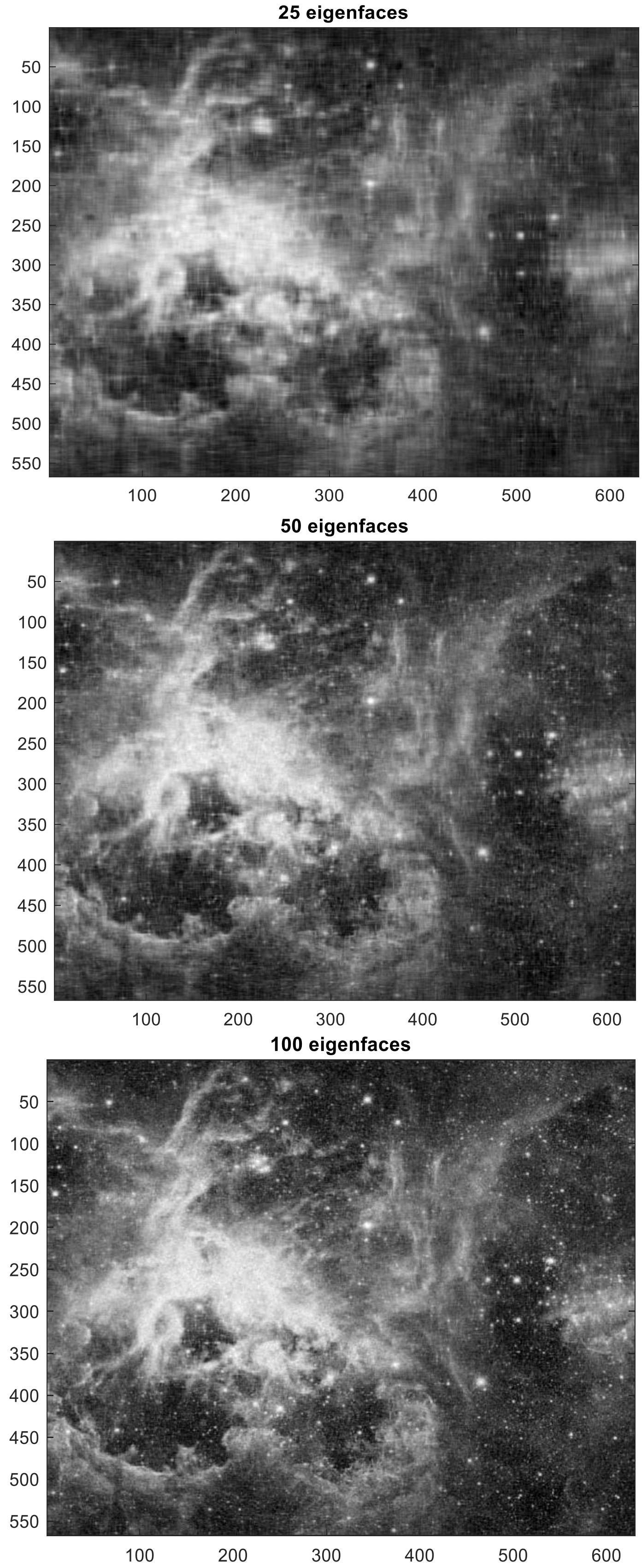


Figure 4,5,6- Nasa images that include 25, 50 and 100 eigenfaces.

Q2 – SMİLİNG CLASSİFİER

Codes:

```
clc; clear all; close all;

s = load('Yale.mat','fea','gnd');
face=s.fea; label=s.gnd;
s_ind=3:11:157; n_ind=6:11:160;
sn_ind=[s_ind, n_ind]; faces=face(sn_ind,:); %Here, we
load the matrix that contains every images as a vector,
and we extract the smiling and neutral images.

faceW = 32;  faceH = 32;
numFaces=30; numPerLine = 11;  ShowLine = 2;

Y = zeros(faceH*ShowLine,faceW*numPerLine);
for i=0:ShowLine-1
    for j=0:numPerLine-1
        Y(i*faceH+1:(i+1)*faceH,j*faceW+1:(j+1)*faceW) =
reshape(faces(i*numPerLine+j+1,:),[faceH,faceW]);
    end
end
imagesc(Y);colormap(gray); %Here, we plot images in one
figure.

neutral = [16:30];
smile = [1:15];

% Subtract the mean 'face' before performing PCA
h = 32; w = 32;
meanFace = mean(faces, 1);
faces = faces - repmat(meanFace, numFaces, 1);

[u,d,v] = svd(faces.', 'econ'); %we give the 'econ' as the
second parameter to get 'u' matrix that has 1024 rows
eigVals = diag(d);
eigVecs = u; % Pull out eigen values and vectors

% Plot the mean sample and the first three principal
components
figure; imagesc(reshape(meanFace, h, w)); colormap(gray);
title('Mean Face');
figure;
subplot(1, 3, 1); imagesc(reshape(u(:, 1), h, w));
colormap(gray);title('First Eigenface');
```

```
subplot(1, 3, 2); imagesc(reshape(u(:, 2), h, w));
colormap(gray);title('Second Eigenface');
subplot(1, 3, 3); imagesc(reshape(u(:, 3), h, w));
colormap(gray);title('Third Eigenface');

neutralFaces = faces(neutral, :); smileFaces =
faces(smile, :);
neutralWeights = eigVecs(:,16:30) * neutralFaces;
smileWeights = eigVecs(:,1:15) * smileFaces;

for i = 1:length(smile)
test_smile=smileWeights(:,i);
test_repeat_smile=repmat(test_smile,1,(length(smile)-1));
smile_weights_no_test=[smileWeights(:,1:i-1)
smileWeights(:,i+1:end)];
distance_smile=test_repeat_smile-
smile_weights_no_test(:,1:14);
distance_smile_val(i)=sum(vecnorm(distance_smile))/(length
(smile)-1);
end

for i = 1:length(smile)
test_smile=smileWeights(:,i);
test_repeat_neutral=repmat(test_smile,1,(length(neutral)))
;
distance_neutral=test_repeat_neutral-
neutralWeights(:,1:15);
distance_neutral_val(i)=sum(vecnorm(distance_neutral))/(le
ngth(neutral));
end

for i = 1:length(smile)
   decision(i)=distance_neutral_val(i)>=
distance_smile_val(i) %if this  condition is true,
desicion(i) will equal to 1 and label smiling class
end
sum(decision)/length(smile) % it gives a score that tells
us how accurately we can predict the smiling class.
```

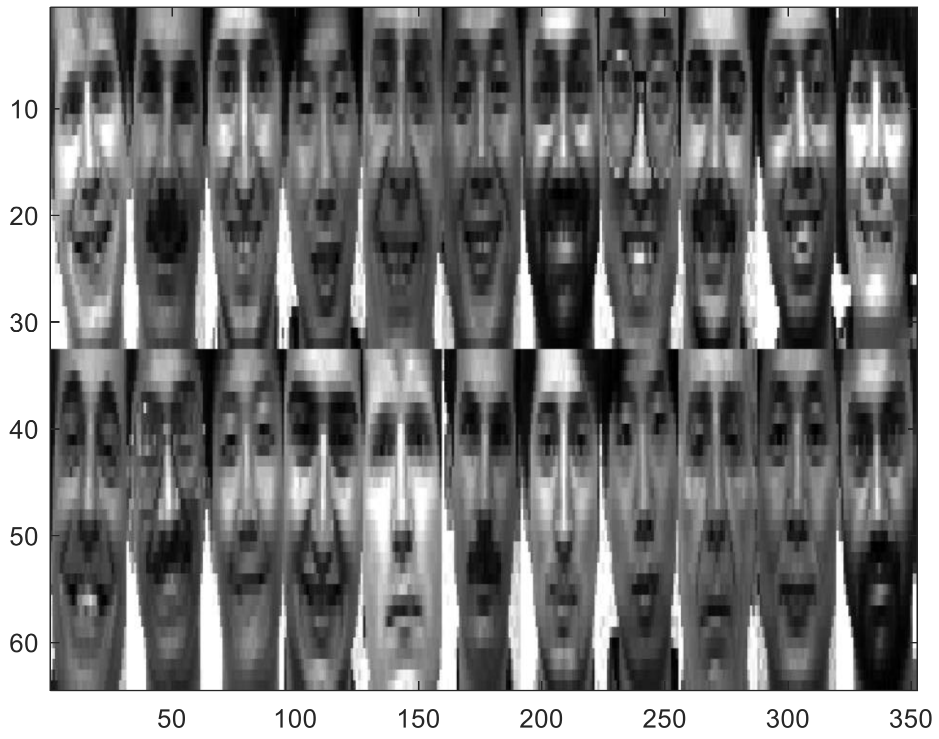


Figure 1- Smiling and neutral faces images in same figure.

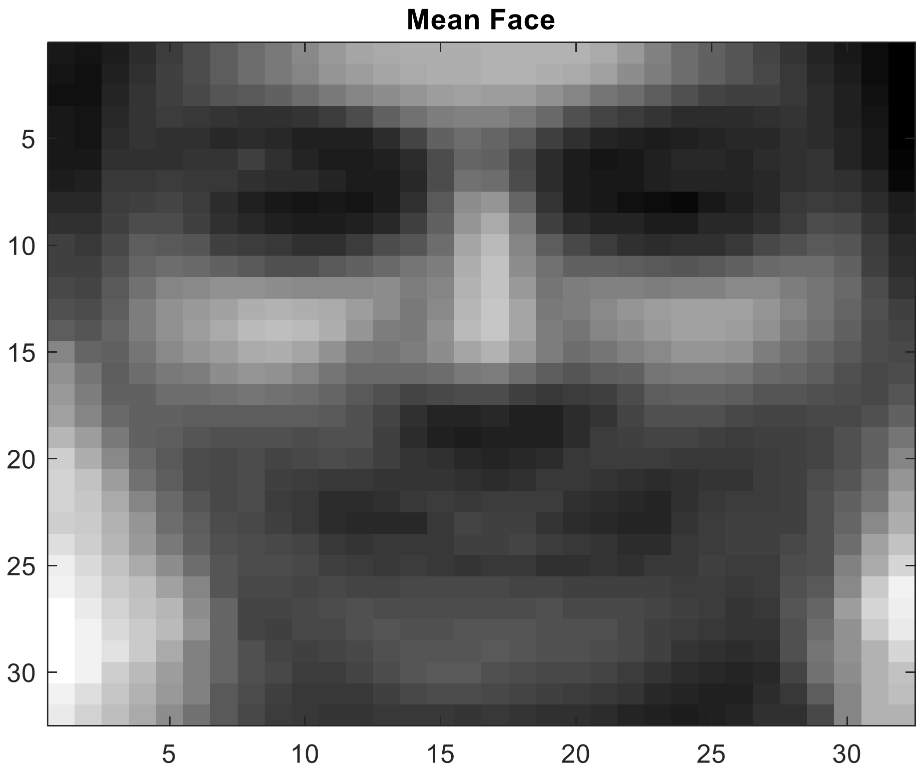


Figure 2 – Mean Face that we get

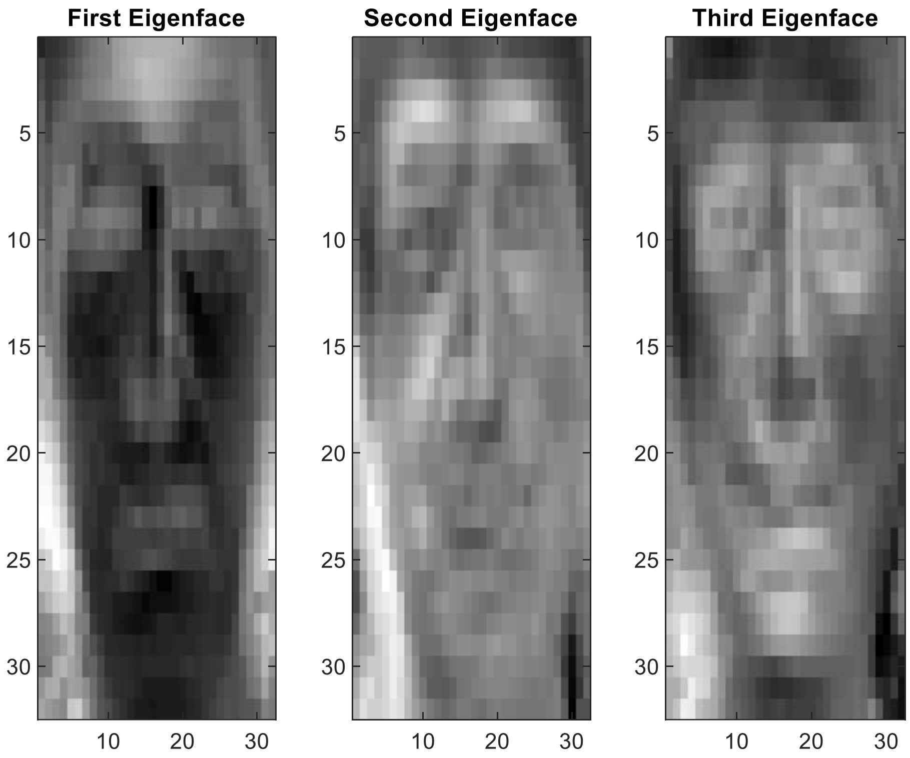
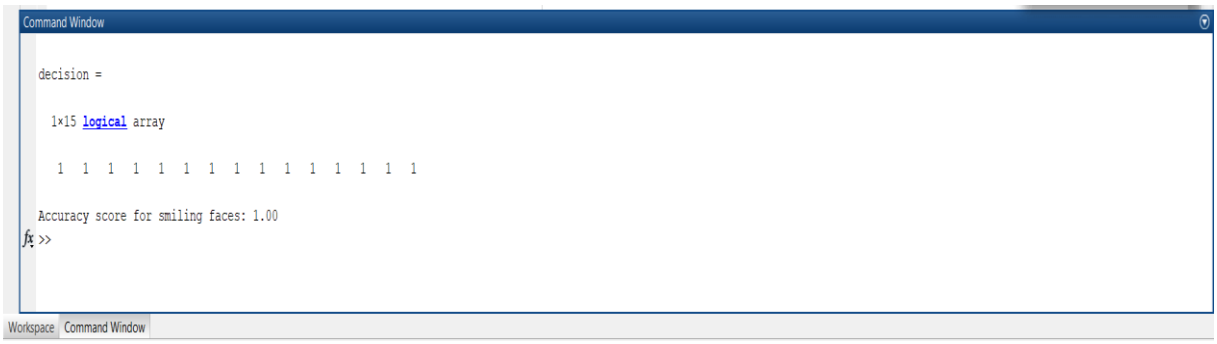


Figure 3 – First, Second and Third Eigenfaces



```
decision =

  1×15 logical array

   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1

Accuracy score for smiling faces: 1.00
fx >>
```

Figure 4 – Acuracy Score and Decision Vector

As we see on command window, We classify accurately all smiling faces.

Q3 – GENDER CLASSİFİER

Codes:

```
clc;clear all;close all;

menFolder = 'faces_men'; listname1 =
dir(fullfile(menFolder,'*.jpg'));
men=[];
for k = 1:length(listname1)
    man=reshape(imread([menFolder filesep
listname1(k).name]),[1,1296]);
    men=[men; man];
end

womenFolder = 'faces_women'; listname2 =
dir(fullfile(womenFolder,'*.jpg'));
women=[];
for k = 1:length(listname2)
    woman=reshape(imread([womenFolder filesep
listname2(k).name]),[1,1296]);
    women=[women; woman];
end
faces=double([men; women]);
%We read image files as a vector and store these in faces matrix

faceW = 36;  faceH = 36;
numFaces=400; numPerLine = 10;  ShowLine = 2;

Y = zeros(faceH*ShowLine,faceW*numPerLine);
Z = zeros(faceH*ShowLine,faceW*numPerLine);
for i=0:ShowLine-1
    for j=0:numPerLine-1
        Y(i*faceH+1:(i+1)*faceH,j*faceW+1:(j+1)*faceW) =
reshape(men(i*numPerLine+j+1,:),[faceH,faceW]);
        Z(i*faceH+1:(i+1)*faceH,j*faceW+1:(j+1)*faceW) =
reshape(women(i*numPerLine+j+1,:),[faceH,faceW]);
    end
end
subplot(2,1,1); imagesc(Y);colormap(gray);
subplot(2,1,2);imagesc(Z);colormap(gray); %Here, we plot images in
one figure.

men = [1:200];  women = [200:400];
% Subtract the mean 'face' before performing PCA
h = 36; w = 36;
meanFace = mean(faces, 1);
faces = faces - repmat(meanFace, numFaces, 1);

[u,d,v] = svd(faces.', 'econ'); %we give the 'econ' as the second
parameter to get 'u' matrix that has 1024 rows
eigVals = diag(d);
eigVecs = u; % Pull out eigen values and vectors
```

```
% Plot the mean sample and the first three principal components
figure; imagesc(reshape(meanFace, h, w)); colormap(gray);
title('Mean Face');
figure;
subplot(1, 3, 1); imagesc(reshape(u(:, 1), h, w));
colormap(gray);title('First Eigenface');
subplot(1, 3, 2); imagesc(reshape(u(:, 2), h, w));
colormap(gray);title('Second Eigenface');
subplot(1, 3, 3); imagesc(reshape(u(:, 3), h, w));
colormap(gray);title('Third Eigenface');

womenFaces = faces(women, :); menFaces = faces(men, :);
womenWeights = eigVecs(:,200:400) * womenFaces;
menWeights = eigVecs(:,1:200) * menFaces;

for i = 1:length(men)
    test_men=menWeights(:,i);
    test_repeat_men=repmat(test_men,1,(length(men)-1));
    men_weights_no_test=[menWeights(:,1:i-1) menWeights(:,i+1:end)];
    distance_men=test_repeat_men-men_weights_no_test(:,length(men)-
1);
    distance_men_val(i)=sum(vecnorm(distance_men))/(length(men)-1);
end

for i = 1:length(men)
    test_men=menWeights(:,i);
    test_repeat_women=repmat(test_men,1,(length(women)));
    distance_women=test_repeat_women-
womenWeights(:,1:length(men)+1);

distance_women_val(i)=sum(vecnorm(distance_women))/(length(women));
end

for i = 1:length(men)
   decision(i)=distance_women_val(i)>= distance_men_val(i); %if this
condition is true, desicion(i) will equal to 1 and label men class
end
accuracy=sum(decision)/length(men); % it gives a score that tells us
how accurately we can predict the smiling class.

decision
fprintf("Accuracy score for men faces: %.2f\n",accuracy);
```

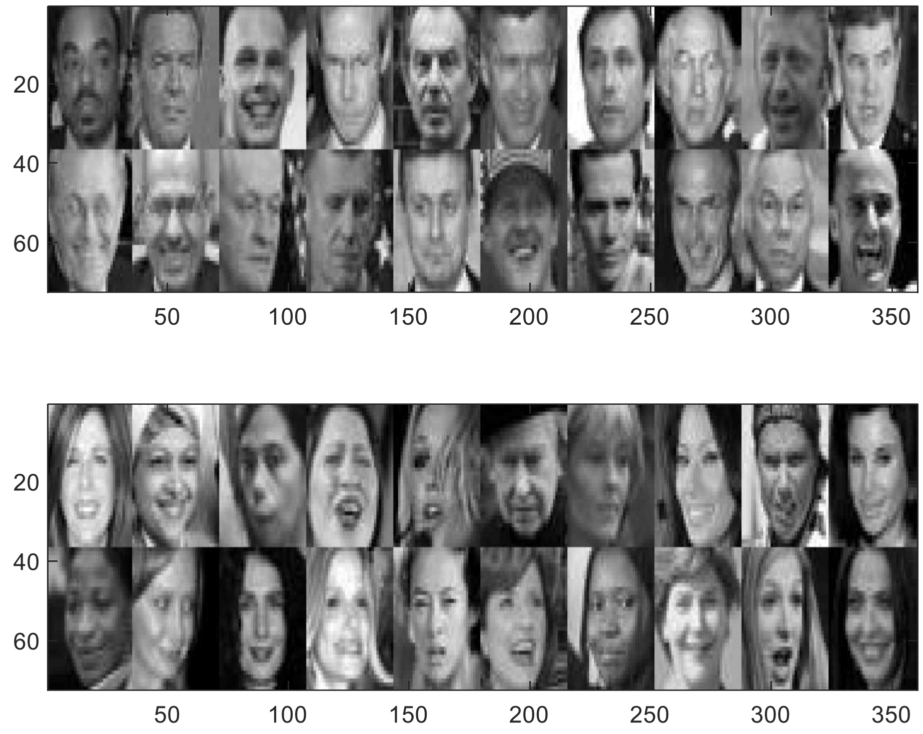


Figure 1- Men and Women faces images in same figure.

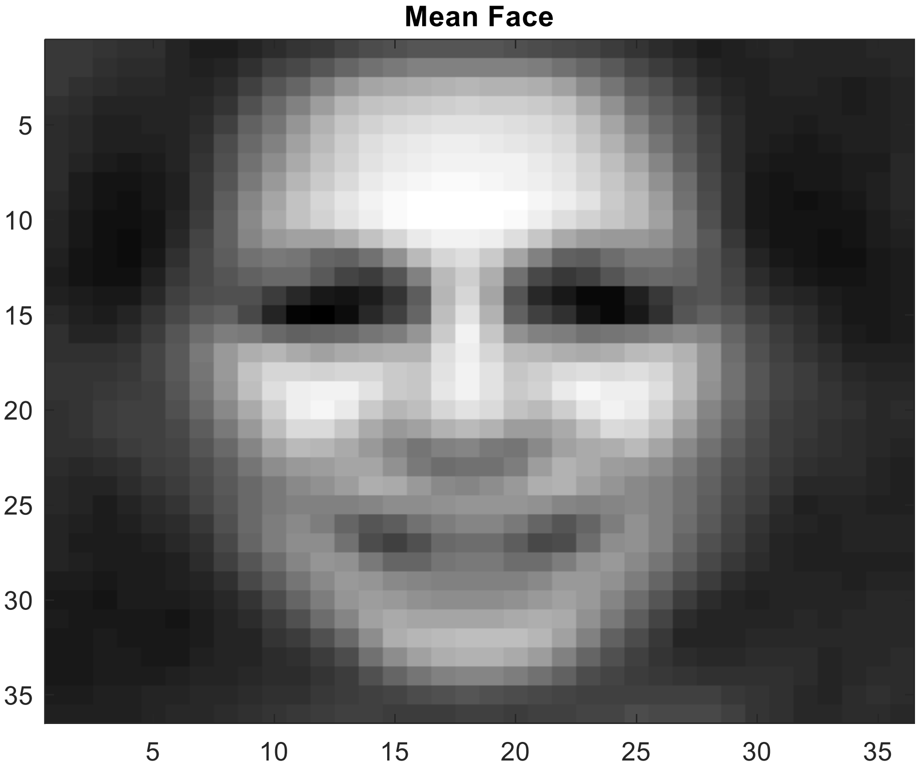


Figure 2 – Mean Face that we get

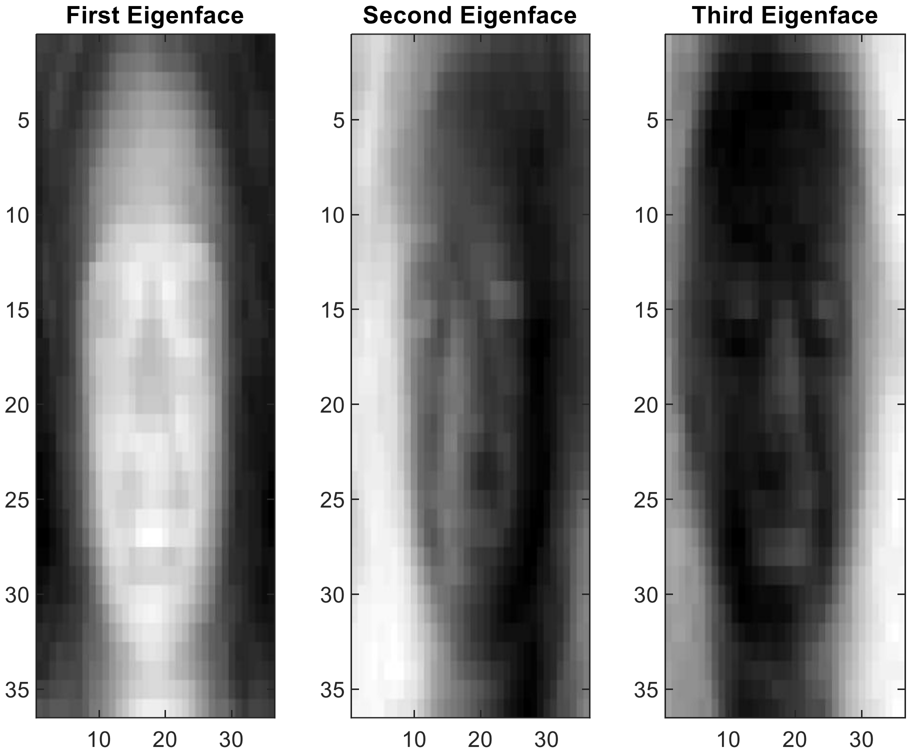
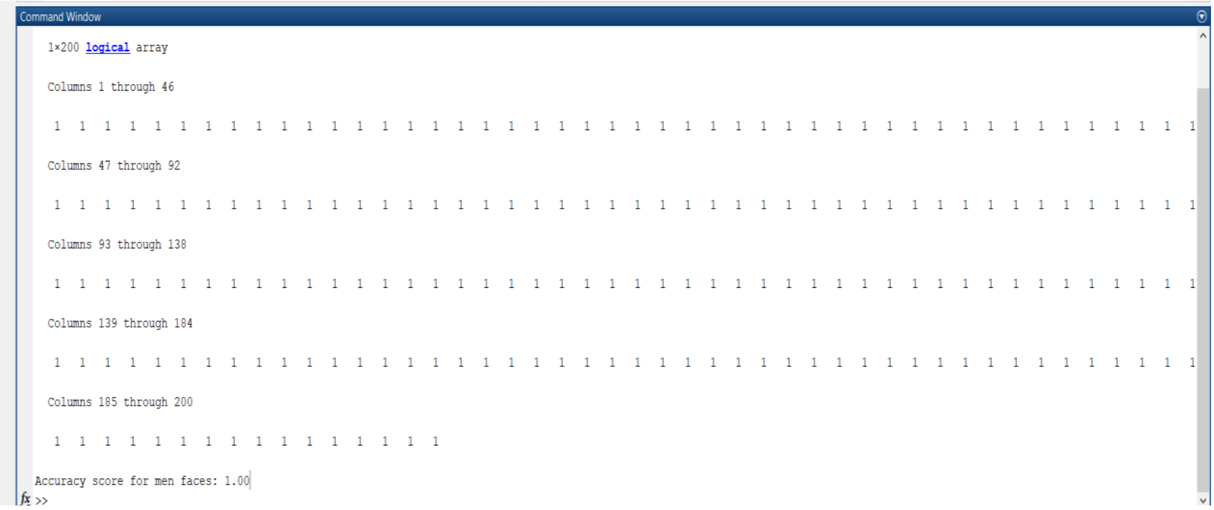


Figure 3 – First, Second and Third Eigenfaces



```
  1×200 logical array

  Columns 1 through 46

   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1

  Columns 47 through 92

   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1

  Columns 93 through 138

   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1

  Columns 139 through 184

   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1

  Columns 185 through 200

   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1

Accuracy score for men faces: 1.00
fx >>
```

Figure 4 – Acuracy Score and Decision Vector

As we see on command window, We classify accurately all men faces.