Mini Project2 for ECE269A

Yunhai Han
A53307224
y8han@eng.ucsd.edu

1 part(a)

1.1 Restatement of the problem

In part(a), I am required to computer the principal components(PCs) using the first 190 individuals' neutral expression image. Plot the singular values of the data matrix and justify your choice of principal components.

1.2 Results

Here, I don't copy the mathematics derivations from the given paper *Face Recognition Using Eigenfaces* and just list the results the explanations.

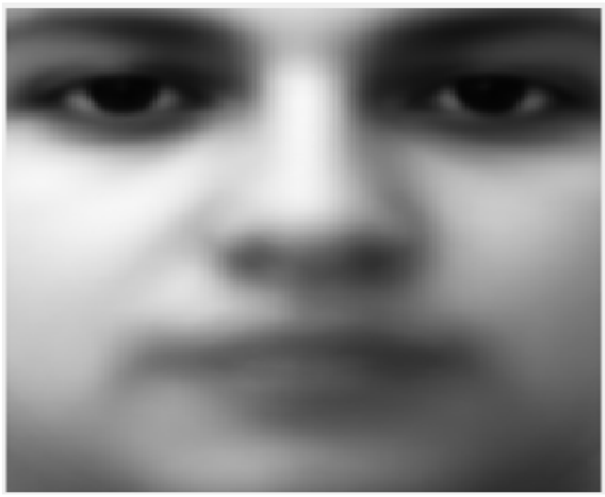
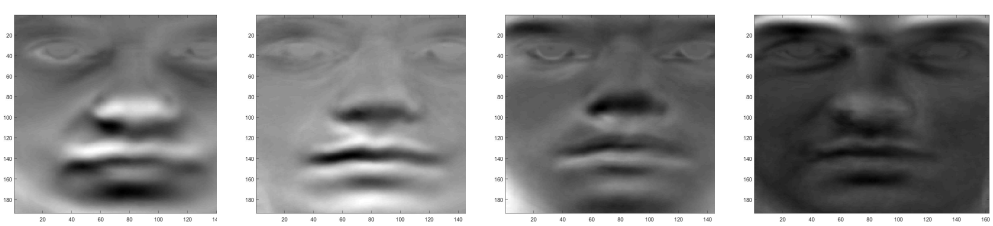


Figure 1: The average face



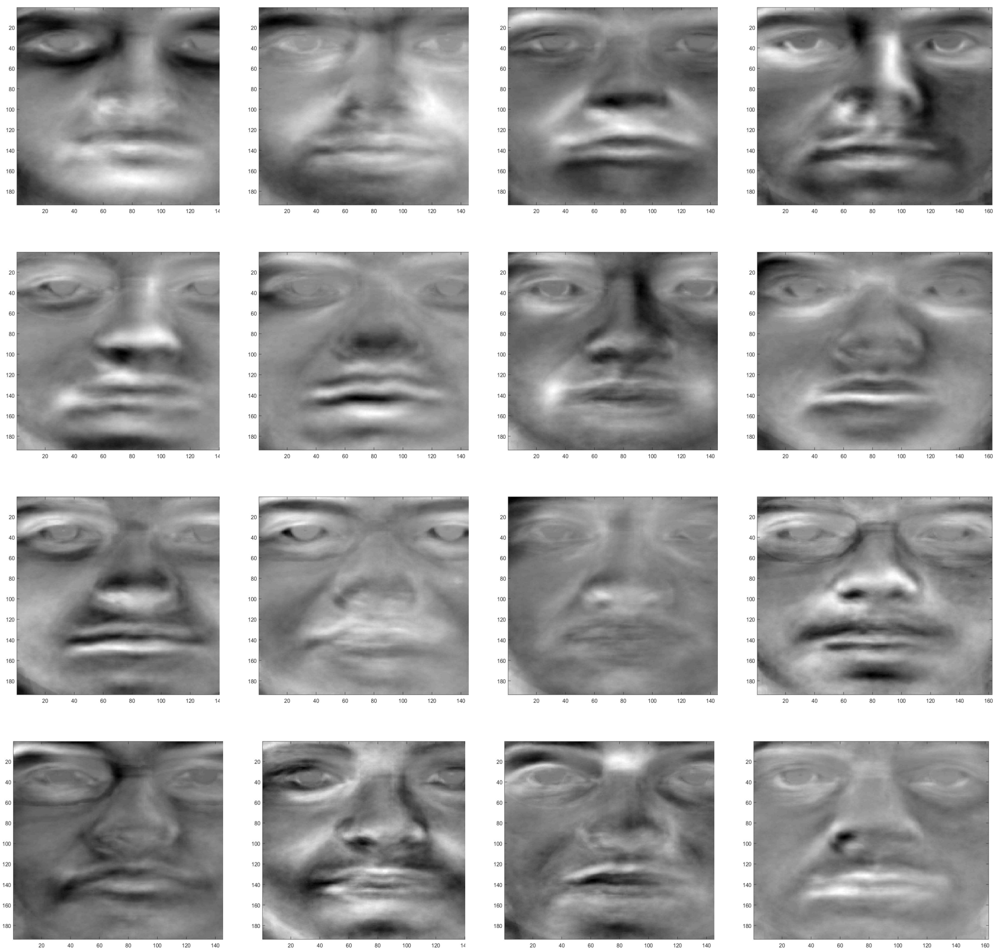


Figure 2: 20 different eigenfaces

From Figure 9, you can see there are 20 different eiganfaces and they are in the descent order with their eigenvalues, which means the first eigenface corresponds to the largest eigenvalue. It indeed there are 190 different eigenfaes because I am now using 190 individuals' neutral expression image. However, as the paper suggests, fewer number of eigenfaces with the largest eigenvalues contain enough information for the following reconstruction, I decide to choose the 20 largest eigenvalues and plot the eigenfaces. Because the eigenface(eigenvector) is the direction of the greatest variance and the corresponding eigenvalues is the variance along that direction. It allows me to use eigenface to describe the distribution of all the face images. Each individual face could be represented exactly in the terms of a linear combination of the eigenfaces. For the computational coefficiency, I choose the first 20 eigenfaces, if the error is beyond the threshold, I would change the number.

2 part(b)

2.1 Restatement of the problem

In part(b), I am required to reconstruct one of 190 individuals' neutral expression image using different number of PCs. And as I vary the number of PCs, I need to plot the plot the mean squared error(MSE) of reconstruction versus the number of principal components to show the accuracy of reconstruction.

2.2 Results

I choose the 13^{th} individual's neutral expression.

In order to reconstruct the expression using different eigenfaces, first I need to project the mean-adjusted face image onto the subspace spanned by the eigenfaces. After this, I could obtain the weights for each eigenface and then, a linear combination of each eigenface with corresponding weight reconstruct the original face image.

First, I use 20 different eigenfaces and reconstruct the expression.

Figure 3: The original expression

Figure 4: Reconstructed expression

The left figure shows the original face expression from the dataset, and the right one shows the reconstructed face expression using 20 different eigenfaces.

I also need to calculate the mean squared error(MSE), and its definition is:

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

In equation 1, n represents the number of all elements in the image, which is euqal to the multiplication of the number of rows and number of columns. Y_i represents each element in the original image and $\hat{Y}_i$ represents each element in the reconstructed image.

In this case(20 different eigenfaces), the mean squared error is 1460.1 and from the two figures, I could tell the reconstructed image is similar with the original one. And that could prove the algorithm works!

Then, I vary the number of PCs, and plot the MSE versus of the number of PCs.

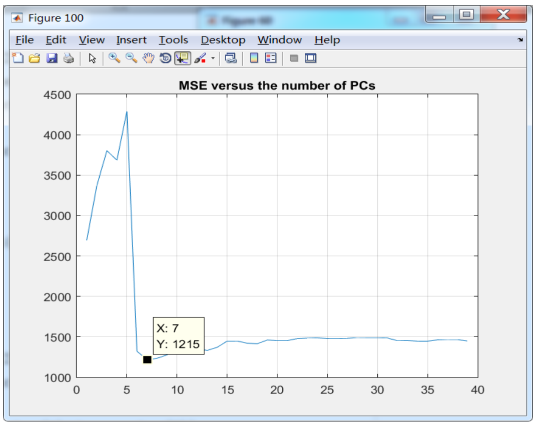


Figure 5: MSE versus the number of PCs

From figure 5, I find when the number is 7, the mean squared error is the smallest(1215). And the paper tells us that only 7 eigenfaces are enough for the reconstruction, which agrees with the results. The more eigenfaces(more than 7)for reconstruction, the larger mean squared error occurs, which is undesirable. Also, if the number of PCs is smaller than 7, the information are not enough to describe the distribution of these face images, which causes very large errors. When the number is 5, the error is the largest(4290).

I put the reconstructed face images using 5 eigenfaces and 7 eigenfaces respectively.

Figure 6: With 5 eigenfaces

Figure 7: With 7 eigenfaces

In figure 6, you could see that the reconstructed face is different from the original one, which also proves the information in only 5 eigenfaces are not enough. In figure 7, you could see that the reconstructed face is similar with another reconstructed face shown in figure 4, which means the accuracy for reconstruction would not change a lot for any number of PCs larger than 7. Besides, figure 5 also proves this conclusion because the MSE nearly remains the same when the number of PCs is larger than 7.

3 part(c)

3.1 Restatement of the problem

In part(c), I am required to reconstruct one of 190 individuals' smiling expression image using different number of PCs. And as I vary the number of PCs, I need to plot the plot the mean squared error(MSE) of reconstruction versus the number of principal components to show the accuracy of reconstruction.

3.2 Results

The only thing I need to do is to change the face expression images for the creation of eigenfaces. With Matlab codes, I could handle this simply by changing the files' name.

Here, I put all the results as in part(a) and part(b).

Figure 8: The average face(similing)

Figure 9: 20 different eigenfaces(smiling)

As in part(b), I need to reconstruct one of 190 individuals' smiling expression images. Here I also choose the 13^{th} individual's smiling expression.

First, I use 20 different eigenfaces for reconstruction.

Figure 10: The original expression

Figure 11: Reconstructed expression

Then, I vary the number of PCs, and plot the MSE versus of the number of PCs.

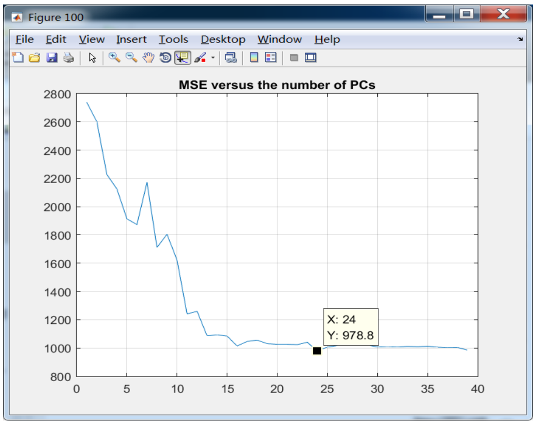


Figure 12: MSE versus the number of PCs

From figure 12, I find when the number is 24, the mean squared error is the smallest(978.8).However, it seems to me that it is more difficult for the reconstruction of individuals' smiling expression because the curve is not as ideal as it in figure 5. Also, if the number of PCs is smaller than 15, the information are not enough to describe the distribution of these face images, which causes very large errors. When the number is 2, the error is the largest(2601).

I put the reconstructed face images using 2 eigenfaces and 24 eigenfaces respectively.

Figure 13: With 2 eigenfaces

Figure 14: With 24 eigenfaces

In figure 16, you could see that the reconstructed face is different from the original one, which also proves the information in only 2 eigenfaces are not enough. In figure 14, you could see that the reconstructed face is similar with another reconstructed face shown in figure 11, which means the accuracy for reconstruction would not change a lot for any number of PCs larger than 20. Besides, figure 12 also proves this conclusion because the MSE nearly remains the same when the number of PCs is larger than 20.

4 part(d)

4.1 Restatement of the problem

In part(d), I am required to reconstruct one of the other 10 individuals' neutral expression image using different number of PCs. And as I vary the number of PCs, I need to plot the mean squared error(MSE) of reconstruction versus the number of principal components to show the accuracy of reconstruction.

4.2 Results

I choose the '198a.jpg' and plot it here.

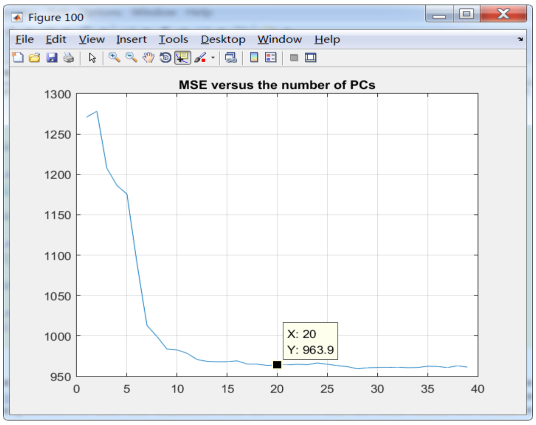


Figure 15: MSE versus the number of PCs

From figure 21, you could see the MSE is smallest(963.9) when the number of PCs is about 20. And when the number is larger than 20, the MSE nearly remains the same.

Figure 16: 198a

Figure 17: 2 eigenfaces

Figure 18: 20 eigenfaces

From figure 17 and figure 18, I could tell that the two reconstructed faces look very similar. The could be validated by the errors shown in figure 21 because the difference between each error is smaller than these in the previous parts. You can see that when the number is 2, the MSE is the largest(1278) and the smallest one is 963.9. So, in this case, we could even use 2 or 3 eigenfaces to reconstruct the original face expression.

5 part(e)

5.1 Restatement of the problem

In part(e), I am required to use any other non-human image(e.g.,car image, resize and crop to the same size),and try to reconstruct it using all the PCs.

5.2 Results

I choose to use a car image and resize it.

Figure 19: A nice car

The problem require me to use all the PCs, so I don't have to decide the number of PCs, I use all of them(190).

Figure 20: Reconstructed face from a car image

In figure 20, it looks like a face(and it even looks like the lady in figure 16). However, the mean squared error is 6710.5 and based on this, I could still make a decision that the original image contains no human face(In the paper, this situation is called 'distant from face space')

In order to examine whether this conclusion is true or not, I use another non-human image(building).

Figure 21: A tall building

Also, I use all the PCs.

Figure 22: Reconstructed face from a building image

The mean square error is 3156.5, which is also mush larger then these in the previous parts.

Moreover, in the paper, the author says that many images(most of them looking nothing like a face) will project onto a given pattern vector and this is not a problem for the system.

From figure 20 and figure 22, you can see that the two reconstructed face look similar. It agrees with what the author says in the paper since both of them are non-human images and thus, they project onto the same pattern vector.

Hence, I could set a threshold to distinguish human face images and other non-human images. If just considering the chosen faces, the smallest MSE are 1215, 978.8, 963.9, so if I set the threshold to be 2000, it could distinguish the building and the car from human faces.

In practice, the threshold should be obtained after many tests using different images and the selected threshold could distinguish the most images from human faces successfully. However, if it is not a linear classifier, there always exist a few images which could not be distinguished from human faces(the MSE are smaller than the selected threshold but it is a non-human image).

6 part(f)

6.1 Restatement of the problem

In part(f), I am required to rotate one of the 190 individuals' neutral expression image with different degrees and try to reconstruct it using all PCs.

6.2 Results

As in part(b) and part(c), I choose the 13^{th} individual's neutral expression and rotate it with different degrees from $1°$ to $60°$. For each degree, I calculate the MSE and keep them in a vector for further analysis.

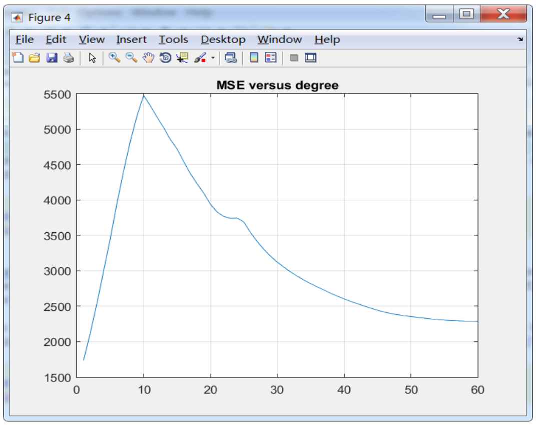


Figure 23: MSE versus degree

For every $5°$, I output the reconstructed face image.

Figure 24: $5°$

Figure 25: $10°$

Figure 26: $15°$

Figure 27: $20°$

Figure 28: $25°$

Figure 29: $30°$

Figure 30: $35°$

Figure 31: $40°$

Figure 32: $45°$

Figure 33: $50°$

Figure 34: $55°$

Figure 35: $60°$

From figure 23, it's hard to tell the relationship between MSE and degrees(MSE increases at first and then decreases when the degree is larger than $10°$). Because of this, MSE only is not enough for us to handle with the classification problem, since they would change with other reasons(rotation in this case).

In the paper, the author also proposes another method for determining which face class provides the best description of an input face image and that is to find the face class k that minimizes the Euclidian distance between the weights vector $\Omega^T = [\omega_1, \omega_2 ... \omega_M]$(M=190).

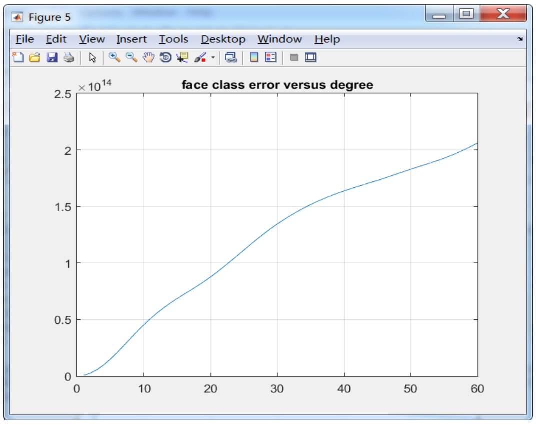


Figure 36: face class error versus degree

From figure 36, you could see there is a great linear relationship between face class error and rotation degree. Based on this, I could design a linear classifier. For example, if the rotation degree is smaller than $10°$, the face class error is smaller than $4.532 * 10^{13}$ and it could still be classified as a face expression. In order to examine whether it works or not, I input the car and building image again and calculate the class face error between them and human-face in '13a.jpg'.

The errors are $3.498 * 10^{14}$ and $1.91 * 10^{14}$ and both are much larger than $4.532 * 10^{13}$, which means MSE and class face error together could successfully distinguish face images and non-human images.

Finally, I try another face image '15a.jpg' and obtain the same results.

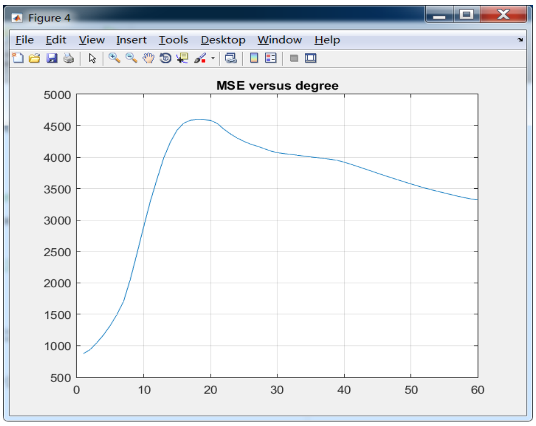


Figure 37: MSE versus degree

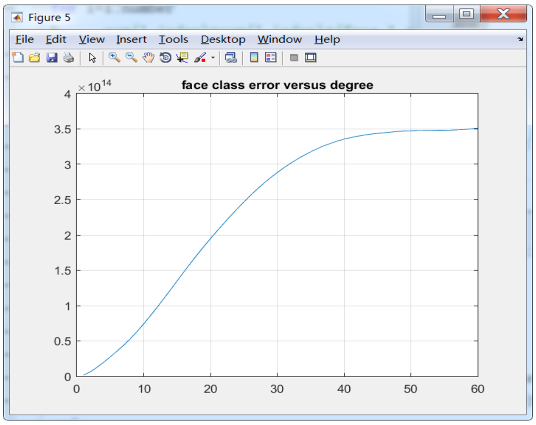


Figure 38: face class error versus degree

The two above figures shows the same trends how MSE and face class error change with degree increasing as before. And the face class errors for car and building images are $4.866*10^{14}$ and $3.407 * 10^{14}$ and they are much larger than $7.401 * 10^{13}$(rotation degree:$10°$).

I put all the Matlab codes here.

Matlab Code

```
%Project for ECE269A
clear;
clc;
close all
m=1;
dir1 = 'C:\Users\Administrator\Desktop\ece269_project\part1\';
for k=1:100
    fn=strcat(dir1,int2str(k),'a.jpg');
    f{m}=imread(fn);
    m=m+1;
end
dir2 = 'C:\Users\Administrator\Desktop\ece269_project\part2\';
for k=101:190
    fn=strcat(dir2,int2str(k),'a.jpg');
    f{m}=imread(fn);
    m=m+1;
end
m=m-1;
image_size=size(f{1});
N2=image_size(1)*image_size(2);
A=zeros(N2,m);
for M=1:m
    A(:,M)=ece_269_project_A(f{M});
end
%A=uint8(A);
aver=sum(A,2)./m;
%aver=uint8(aver);
aver=round(aver);
figure(1);
aver1=ece_269_project_B(aver,image_size);
aver1=uint8(aver1);
imshow(aver1);
saveas(figure(1),['C:\Users\Administrator\Desktop\ece269_project\Solution\figures\aver.jpg']);
```

```
title('The average face')
for i=1:m
    A(:,i)=A(:,i)-aver;
end
C=A'*A;
[eigenvectors,eigenvalues]=eig(C);
eigenfaces=A*eigenvectors;
% sum=zeros(1,40-2+1);
% index=1;
% for number=2:40
% % number=30;
% dir3='C:\Users\Administrator\Desktop\ece269_project\Solution\figures\eigenface';
% for i=1:number
%     eigenfaces_image{i}=ece_269_project_B(eigenfaces(:,191-i),image_size);
% %     figure(i+1)
% %     imagesc(eigenfaces_image{i})
% %     colormap(gray)
% %     fn=strcat(dir3,int2str(i),'b.jpg');
% %     saveas(figure(i+1),fn);
% end
% %then I choose the 13th eigenface for reconstruction
% face_1=ece_269_project_A(f{13});
% face_1_aver=face_1-aver;
% omega=zeros(1,number);
% for i=1:number
%     omega(1,number+1-i)=eigenfaces(:,190+1-i)'*face_1_aver;
% end
% new_face1=zeros(image_size(1)*image_size(2),1);
% for i=1:number
%     new_face1=new_face1+eigenfaces(:,190+1-i)*omega(1,number+1-i);
% end
% new_face2=new_face1;
% new_face22=new_face1;
% face_max=max(new_face1);
% face_min=min(new_face1);
% for i=1:image_size(1)*image_size(2)
%     new_face2(i,1)=round((new_face2(i,1)-face_min)/(face_max-face_min)*255);
%     new_face22(i,1)=round((new_face22(i,1)-face_min)/(face_max-face_min)*255)-127;
% end
% new_face=ece_269_project_B(new_face1,image_size);
% % figure(20)
% % imagesc(new_face);
% % colormap(gray)
% new_face3=uint8(ece_269_project_B(new_face2,image_size));
% figure(20+number)
% %fn=strcat(dir3,'part2b.jpg');
% imshow(new_face3)
% %saveas(figure(21),fn);
% %then I calculate the MSE for different numbers of PCs
% for i=1:image_size(1)*image_size(2)
%     sum(1,index)=sum(1,index)+(face_1_aver(i,1)-new_face22(i,1))^2;
% end
% sum(1,index)=sum(1,index)/(image_size(1)*image_size(2));
% index=index+1;
% end
% figure(100)
% plot(sum);
% grid on
% title('MSE versus the number of PCs')
%this is for part(d)
% sum=zeros(1,40-2+1);  %the number of PCs are 2-40
% index=1;
% for number=2:40
```

```
number_degree=60;
sum_degree=zeros(1,number_degree);
class_degree=zeros(1,number_degree);
for degree=1:number_degree
sum=0;
class_error=0;
number=190;
dir3='C:\Users\Administrator\Desktop\ece269_project\Solution\figures\13a.jpg';
dir5='C:\Users\Administrator\Desktop\ece269_project\Solution\figures\';
% dir4='C:\Users\Administrator\Desktop\ece269_project\Solution\figures\building.jpg
    ';
J=imread(dir3);
J_r=imrotate(J,degree,'bilinear','crop');
imshow(J)
figure(2)
% J_r=imread(dir4);
% J_r=rgb2gray(J_r);
% J_r=imresize(J_r,[193,162]);
% J_r=imrotate(J_r,degree,'bilinear','crop');
imshow(J_r)
for i=1:number
    eigenfaces_image{i}=ece_269_project_B(eigenfaces(:,191-i),image_size);
%     figure(i+1)
%     imagesc(eigenfaces_image{i})
%     colormap(gray)
%     fn=strcat(dir3,int2str(i),'b.jpg');
%     saveas(figure(i+1),fn);
end
%then I choose the 13th eigenface for reconstruction
face_1=ece_269_project_A(J);
face_1_r=ece_269_project_A(J_r);
face_1_aver=face_1-aver;
face_1_aver_r=face_1_r-aver;
omega=zeros(1,number);
omega_r=zeros(1,number);
for i=1:number
    omega(1,number+1-i)=eigenfaces(:,190+1-i)'*face_1_aver;
end
for i=1:number
    omega_r(1,number+1-i)=eigenfaces(:,190+1-i)'*face_1_aver_r;
end
for i=1:number
%     sum(1,index)=sum(1,index)+(face_1_aver(i,1)-new_face22(i,1))^2;
    class_error=class_error+(omega(1,number+1-i)-omega_r(1,number+1-i))^2;
end
% sum(1,index)=sum(1,index)/(image_size(1)*image_size(2));
class_error=class_error/(190);
class_degree(1,degree)=class_error;
new_face1=zeros(image_size(1)*image_size(2),1);
for i=1:number
    new_face1=new_face1+eigenfaces(:,190+1-i)*omega_r(1,number+1-i);
end
new_face2=new_face1;
new_face22=new_face1;
face_max=max(new_face1);
face_min=min(new_face1);
for i=1:image_size(1)*image_size(2)
    new_face2(i,1)=round((new_face2(i,1)-face_min)/(face_max-face_min)*255);
    new_face22(i,1)=round((new_face22(i,1)-face_min)/(face_max-face_min)*255)-127;
end
new_face=ece_269_project_B(new_face1,image_size);
% figure(20)
% imagesc(new_face);
```

```
% colormap(gray)
new_face3=uint8(ece_269_project_B(new_face2,image_size));
if(mod(degree,5)==0)
figure(3+degree)
fn=strcat(dir5,int2str(degree),'part6.jpg');
imshow(new_face3)
saveas(figure(3+degree),fn);
end
%then I calculate the MSE for different numbers of PCs
for i=1:image_size(1)*image_size(2)
%      sum(1,index)=sum(1,index)+(face_1_aver(i,1)-new_face22(i,1))^2;
    sum=sum+(face_1_aver(i,1)-new_face22(i,1))^2;
end
% sum(1,index)=sum(1,index)/(image_size(1)*image_size(2));
sum=sum/(image_size(1)*image_size(2));
sum_degree(1,degree)=sum;
end
figure(4)
plot(sum_degree)
grid on
title('MSE versus degree')
figure(5)
plot(class_degree)
title('face class error versus degree')
%index=index+1;
% end
% figure(100)
% plot(sum);
% grid on
% title('MSE versus the number of PCssu')
```

```
function [ A ] = ece_269_project_A( image )
    image_size=size(image);
    N2=image_size(1)*image_size(2);
    A=zeros(N2,1);
    for i=1:image_size(2)
        A((i-1)*image_size(1)+1:i*image_size(1),1)=image(:,i);
    end
%      A=uint8(A);
end
```

```
function [ A ] = ece_269_project_B( vector,image_size )
    A=zeros(image_size(1),image_size(2));
    for i=1:image_size(2)
        A(:,i)=vector((i-1)*image_size(1)+1:i*image_size(1),1);
    end
%      A=uint8(A);
end
```