Al Albi

CSE 4095

March 29, 2018

Project 1

How to retrieve the files:

- For CSE_Project1_PartExperiments2, always run it first before testing Questions 3, 4, 5, and 6.

For Question 1, I used CSE4095_Project1_Part1_LoadFiles.m and CSE4095_Project1_Part1_ResizeImages. For the Writing Stage, I implemented the functions.

For the Experimental Stage, I used CSE 4095_PartExperiments1 for Question 1. For Question 2, I used CSE 4095_PartExperiments2. For Question 3, I used CSE 4095_PartExperiments3. For Question 4, I used CSE 4095_PartExperiments4, and used VECTOR_ARRAY for eigenfaces and projecting faces. For Question 5 and 6, I also used CSE 4095_PartExperiments4 and adjusted K accordingly. However, I plugged in VECTOR_ARRAY_T for eigen-faces, and VECTOR_ARRAY_T for projecting and recognizing them.

For Question 3, I used CSE 4095_PartExperiments2. I used VECTOR_ARRAY for input for eigenface, reconstructing faces, and projecting faces. There, NUMBER_TEST = 218 and COLNUM = 20. For Questions 4 and 5, I used VECTOR_ARRAY_TEST instead, NUMBER_TEST = 197, and COLNUM = 18.

For parts 3, 4, 5, 6 I always run CSE 4095_PartExperiments2 to show the reconstructions of the faces, adjusting values accordingly.

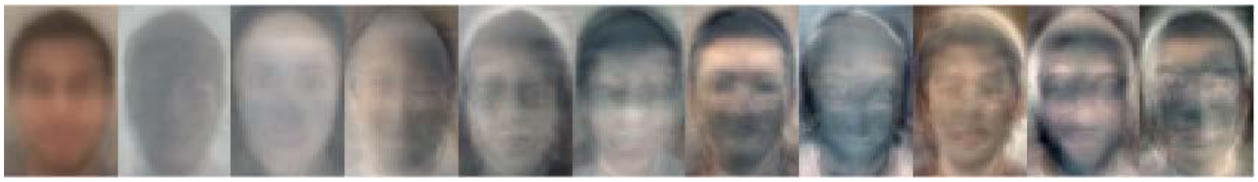
For Question 7, I saved the data in loadfiles.m to plot it in multiple graphs.

Part 1

Question 1

In my program, I implemented the eigenface function. An observation I made in the beginning was that by turning each photo into a vector that is the length of the image's dimensions, no information is lost when computing the covariance. As a result, by having a covariance of 7200 by 7200, we can retrieve the eigen-faces directly from those K vectors. As confirmation of the implementation of this function, the code for this function is in the zip files, and a picture of the average face and K = 10 eigen-faces are shown. The average face is the leftmost.

In the image below, the eigen-faces below are normalized and are not used for the calculation. Their real appearance should be very dark.



Question 2 and Question 3

In my program, I implemented the project_face and the reconstruct_face function, which maps a given image to a set of weights w such that the dot product of w and the eigenfaces results in the original image.

As confirmation, this first command below on the console returns the reconstruction using the 10 weights. The second command returns the original image. The image comparison is on the next page. The image on the left is the original. The weights for this image are also shown. First, I used project_face to map the image to the weights. Then I used those weights in reconstruction_face to return the reconstruction.

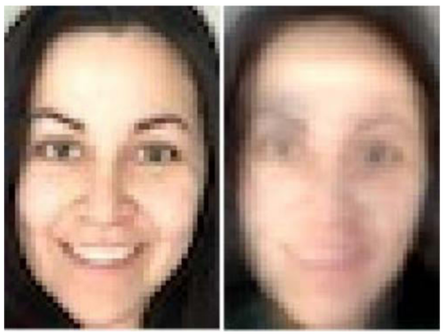
```
imshow(reshape(reconstruct_face(AVERAGEFACE, EIGENFACES, project_face(AVERAGEFACE,
EIGENFACES, VECTOR_ARRAY(:,7))), [60 40 3]));
imshow(reshape(VECTOR_ARRAY(:,7), [60 40 3]));
```

Weights = 7.2828, -12.1758, -5.6878, 4.1303, -8.3178, 1.0698, 3.2777, 3.1912, -0.8988, -3.2200.

Questions 4 and 5

In my program, I used the mean-squared error of the weights to determine how closely related two images are. As confirmation, running the recognize_face function calls on the compare_faces function. Running the command below returns the closest face the program matches it to.

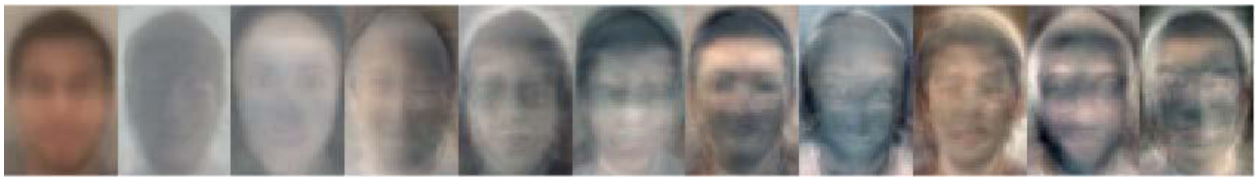
```
[A, B] = recognize_face(AVERAGEFACE, EIGENFACES, FACE_WEIGHTS,
VECTOR_ARRAY(:, 31));
>   imshow(reshape(B, [60 40 3]));
>   imshow(reshape(VECTOR_ARRAY(:, 31), [60 40 3]));
```



Part 2

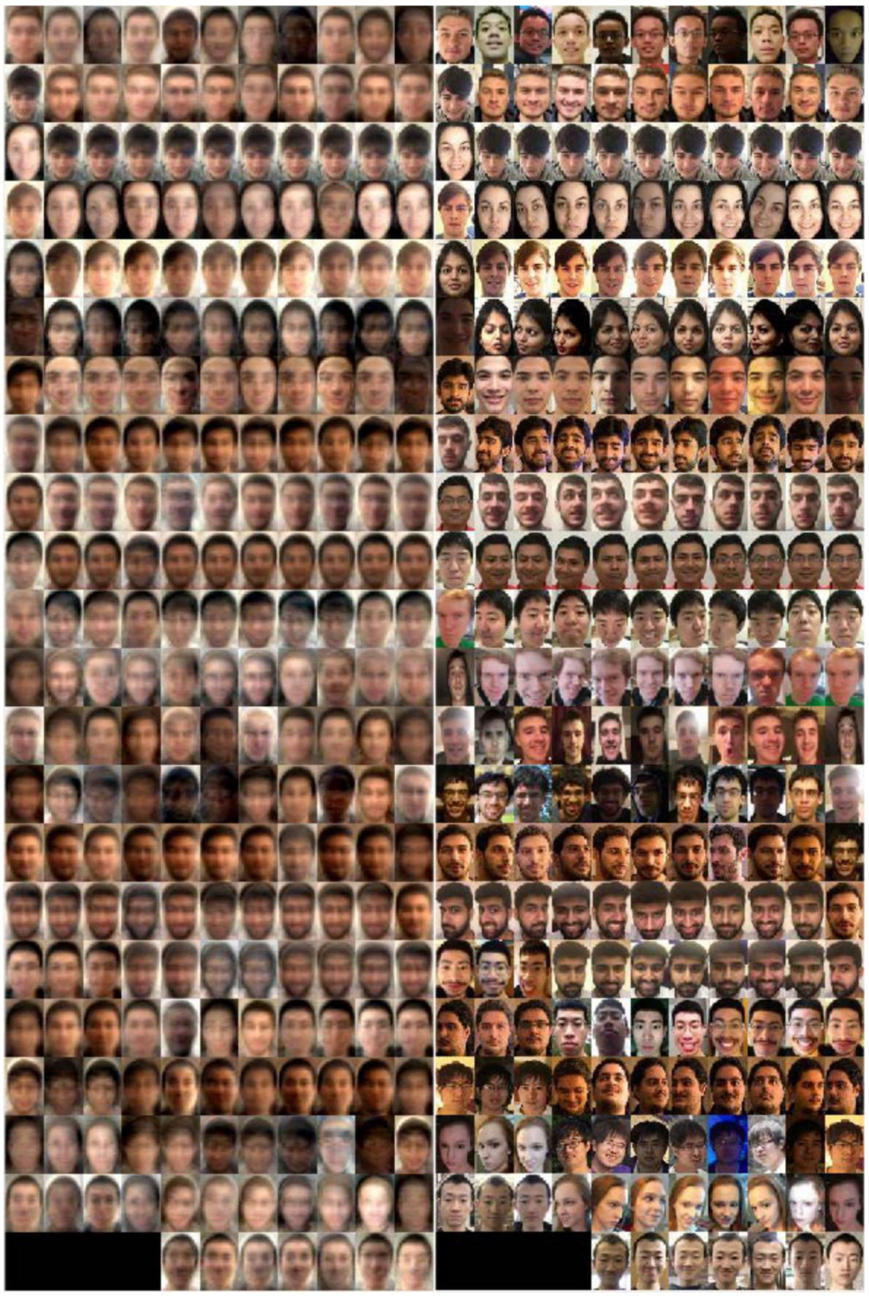
Question 1

For K = 10, below from left to right is the average face, and then the eigen-faces.



Question 2

Below is a reconstruction of each face and their originals.



Question 3

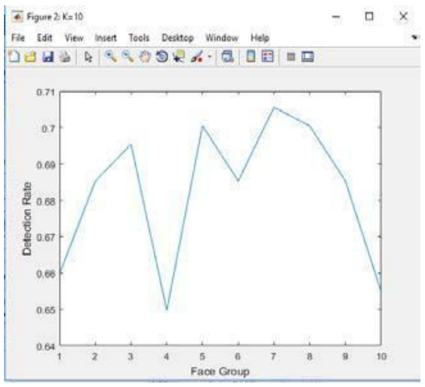
When calculating the detection rate, it was found to be about 75%. In order to test each image against the set without the image itself, for every image in the testing set, the weights corresponding to that image in the testing set were set to zero. Both the training and testing sets were the entire image set.

In order to find the number of successes, the names of the actual images and the approximated images were compared by their first two characters. If they were the same, then add to the counter.

Question 4

For question 4, the code was modified such that the datasets were divided. In particular, since there are 21 persons, the training set only included 21 photos, and the testing set included the total number minus 21. This way, the two sets were mutually exclusive. The training set is named VECTOR_ARRAY_T and the testing set is VECTOR_ARRAY_TEST. Below is a graph of the detection rates by groups, from '01' to '10'.

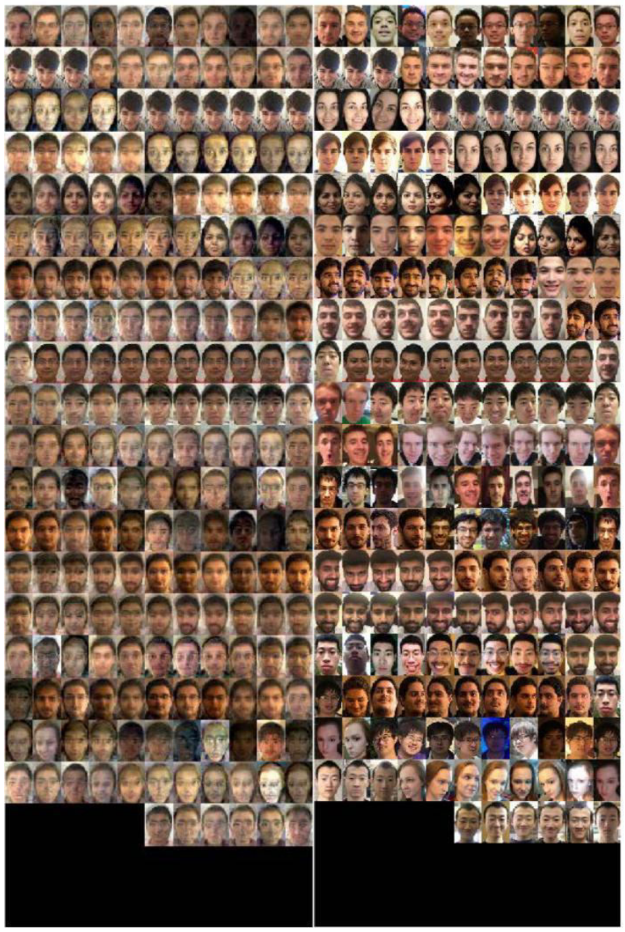
This graph corresponds to the k-th images.

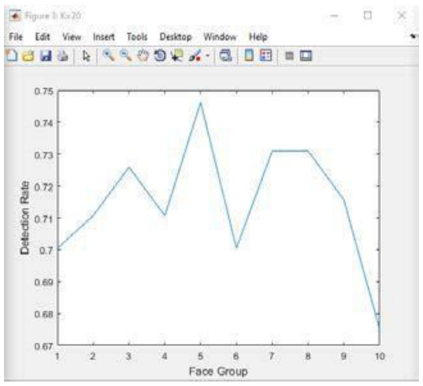


Question 5

For part 5, we modify the code from Question 4 to update K.

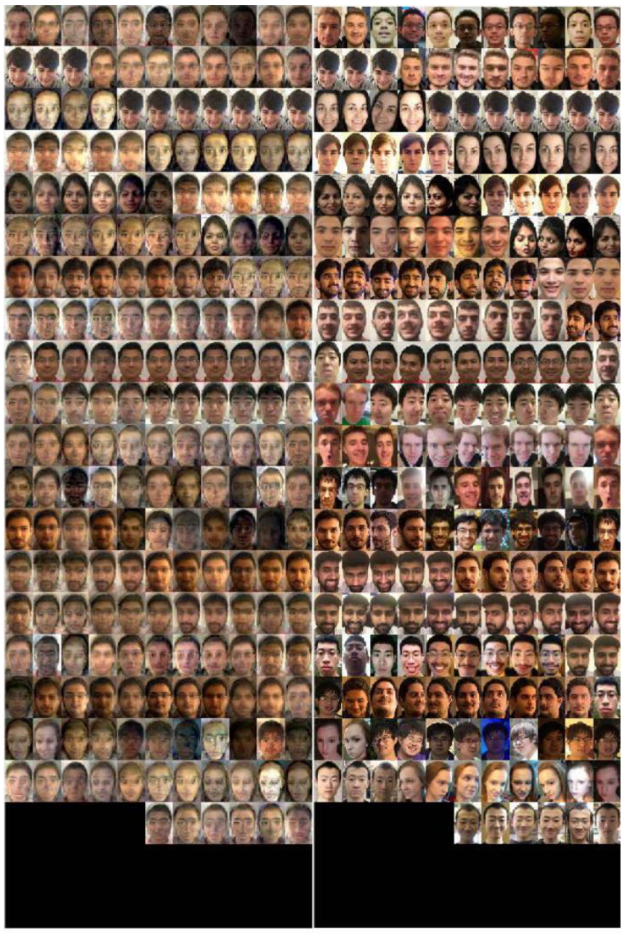
When k = 20, this is the average face and eigenfaces. The average face is determined from the training set.





Question 6

When k = 30, these are the results.



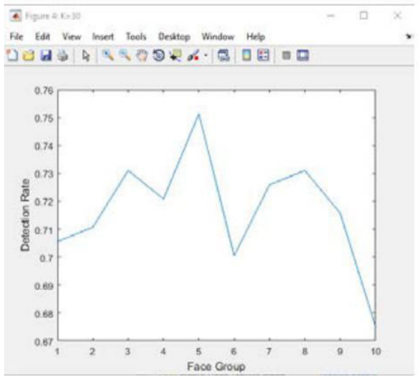
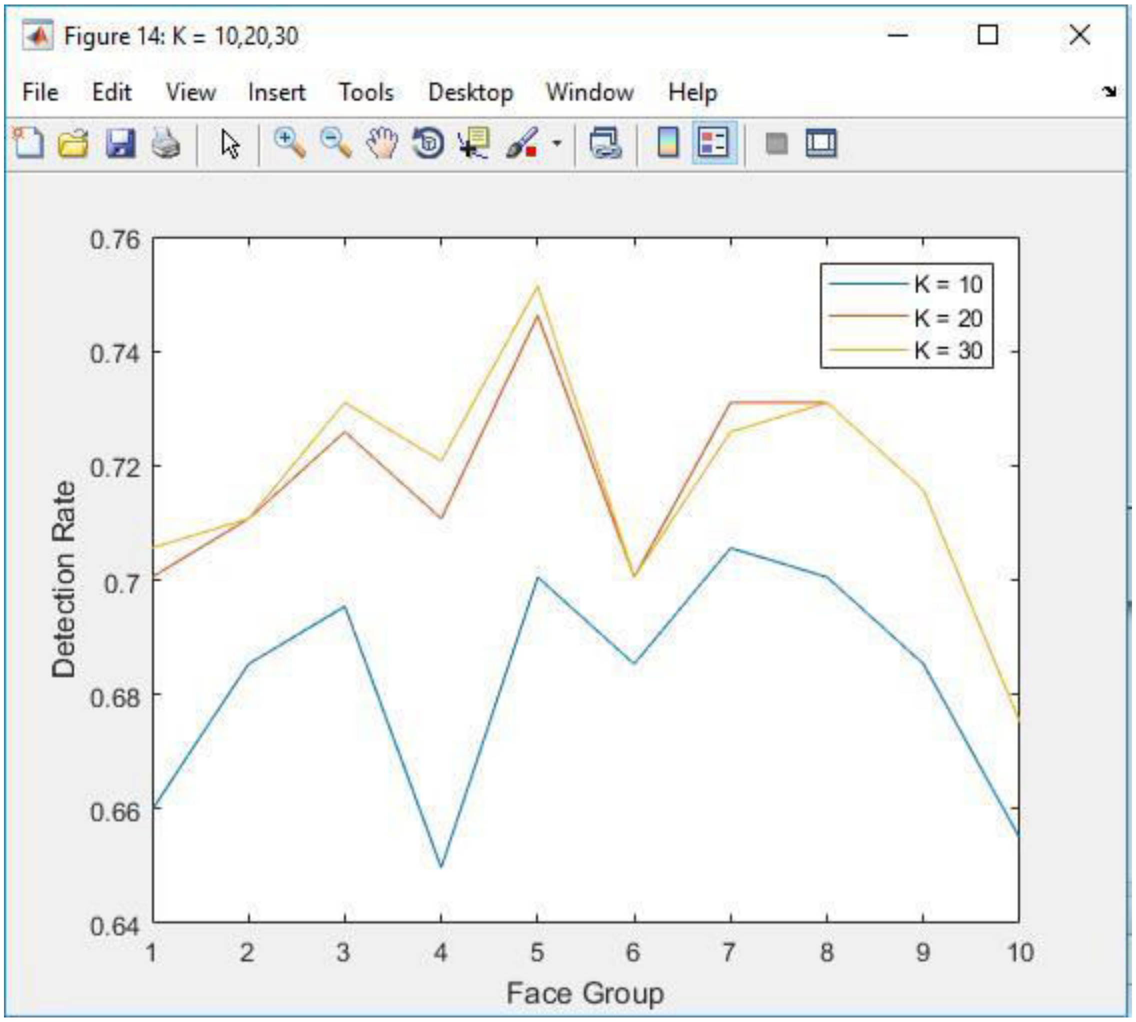


Figure 4: K=30
File Edit View Insert Tools Desktop Window Help
0.76
0.75
0.74
0.73
0.72
0.71
0.7
0.69
0.68
0.67
Detection Rate
1
2
3
4
5
6
7
8
9
10
Face Group

Question 7

The following is the graph of K = 10, 20, 30 with a legend.



Question 8

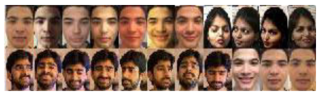
Observing this graph, initially for K = 10, the accuracy was mostly less than 70 percent, with a sharp decline for Face Group 4. This indicates the presence of under-fitting the data with only 10 eigenvalues. When K becomes 20 and then 30, the accuracy converges for each of the Face Groups. Interestingly, for Face Group 7, there also seemed to be signs of overfitting, as K = 20 responded with a higher accuracy than compared to K = 30. This also seemed to be reflected when doing part 2 of the Experiment Stage: although the accuracy was converging, some of the images were also being matched to the wrong faces.

The number of eigenfaces necessary depends on the variance of the dataset. A reliable indicator for choosing the number of eigenfaces is to continue increasing that number until signs of overfitting occur. This happens when increasing the number of eigenfaces actually increases the error, as the program identifies less and less of the test data. Another way to determine the number of eigenfaces is to stop when increasing the number of eigenfaces fails to increase the detection rate by a certain threshold. This will allow the user to specify their goal.

Question 9

When K = 10, the images themselves as observed in Question 2 of the Experimentation stage were recognized with a lower accuracy. The ten eigenvalues were not able to express the detail of the given images significantly. For K = 10, the approximation of the faces appear to resemble the mean face more than the other photos with slight variation.

However, when K = 30, the model begins to over-fit the data and begins to make incorrect assumptions based on the other eigenvalues. Although the over-fitting is not significant in the graph, the over-fitting becomes clearer when observing the images. The recognition error occurs when we have too many eigenvalues because increasing the weights will make the model use finer and less significant features learned from the data that the other models at K = 10 and K = 20 do not pick up. This means that when the images are reconstructed, those smaller details are emphasized and used to determine whether an image is recognized.



Question