Rayan Hassan
4511021

ECE 1395 – Homework 5 – report

Part 1

Here are the accuracies I got for different values of sigma

```
wdir= C:/Users/RAYAN/OneDrive/Desktop )
Reloaded modules: weightedKNN
PART 1

Accuracy with sigma 0.01 is:  0.68
Accuracy with sigma 0.05 is:  0.92
Accuracy with sigma 0.2 is:  0.92
Accuracy with sigma 1.5 is:  0.8
Accuracy with sigma 3.2 is:  0.72
```

This shows that when sigma is too small (close to 0), accuracy decreases (68%). This is because it is as if we have 1 – NN (overfitting). And if sigma is too large, accuracy also starts decreasing as we can see for sigma = 1.5 and sigma = 3.2, since we begin to have under fitting.

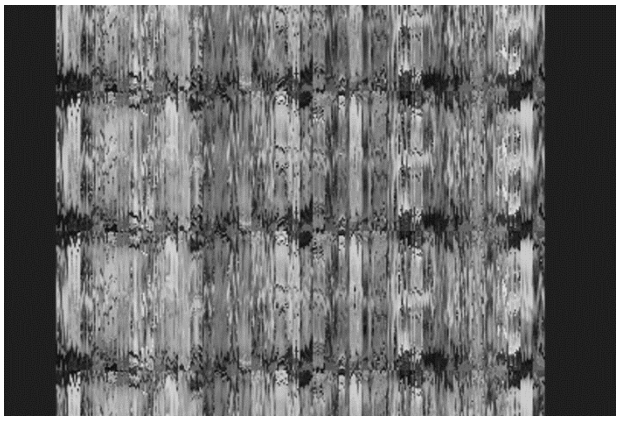
Part 2

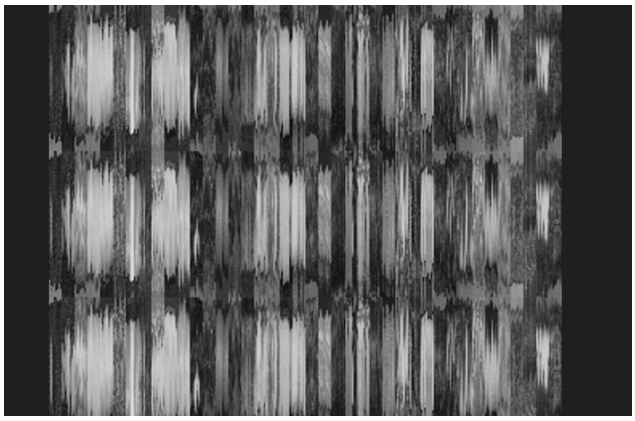
2.0)

Please read the comments I wrote in my code. It should run without any problem, but this just in case. The following is the first picture in my training set (ps5-2-0.png), just to show that I have handle on my data. So it corresponds to person 1

2.1)

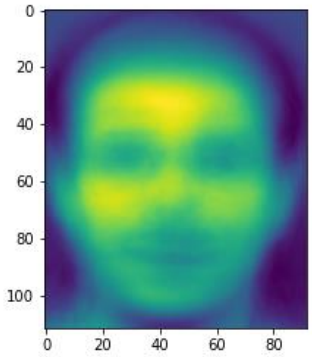
a) Please find different screenshots of the image corresponding to matrix T (ps5-2-1-a.png)





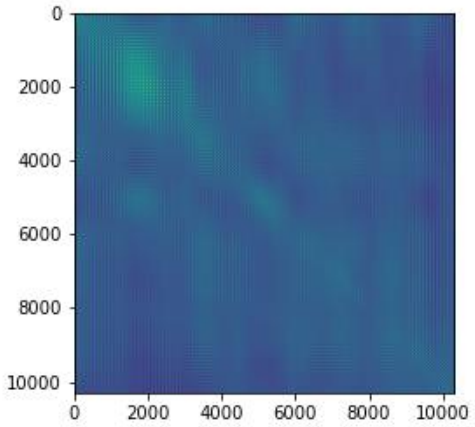
Rayan Hassan
4511021

b) This is the image corresponding to the average face vector (ps5-2-1-b.png)

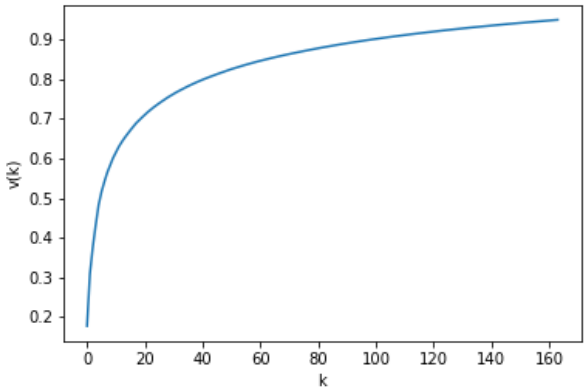


As expected, it looks like a blurry face image. This is because we took the average across each pixel for every image in the training data.

c) This is the image corresponding to the covariance matrix (ps5-2-1-c.png)



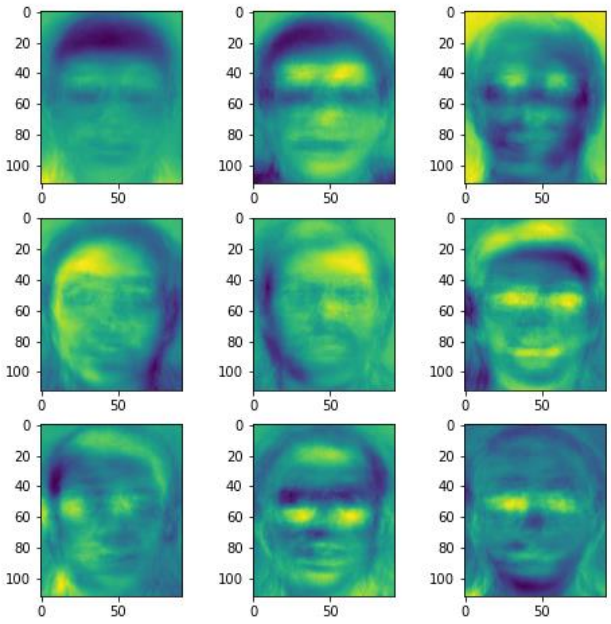
d) This is the plot for k vs v(k). (ps5-2-1-d.png)



The value of k representing the number of eigenvectors that capture 95% of the training data is 163.

Rayan Hassan
4511021

e) The following is an image showing the first 9 eigenfaces.



The dimensions of matrix U are shown below:

```
The dimensions of U are:  (10304, 163)
```

The eigenfaces obtained look like that since each one of them capture different variance of the training data, putting emphasise on some features more than others and vice versa. These are the first 9 eigenfaces, so they capture the most data variance, which is why they all look like faces (i.e we can kind of see the delimitations of the face, eyes, hair, beard, etc.)

2.2)

Please find the dimensions of W_training and W_testing below:

```
W_training dimensions:  (320, 163)
W_testing dimensions:  (80, 163)
```

As we can see, the number of features has been considerably reduced from 10304 to 163.

2.3)

a) The following shows the accuracies for different values of k:

```
Accuracy with k=1:  0.975
Accuracy with k=3:  0.9875
Accuracy with k=5:  0.9125
Accuracy with k=7:  0.8625
Accuracy with k=9:  0.8125
Accuracy with k=11:  0.7875
```

Rayan Hassan
4511021

As excepted, the accuracies are pretty high (higher than 1/n, where n is the number classes)

b) 1) Training time for each model

	One vs One	One vs All
Linear	1.79904 s	0.11993 s
Polynomial	1.70908 s	0.30983 s
RBF	2.00293 s	0.20989 s

2) Testing accuracy for each model

	One vs One	One vs All
Linear	98.75%	100%
Polynomial	88.75%	100%
RBF	100%	98.75%

```
Time for OnevsOne + linear: 1.7990412712097168  s
Accuracy of OnevsOne + linear: 0.9875
```

```
Time for OnevsOne + polynomial: 1.7090859413146973  s
Accuracy of OnevsOne + polynomial: 0.8875
```

```
Time for OnevsOne + gaussian: 2.0029377937316895  s
Accuracy of OnevsOne + gaussian: 1.0
Time for OnevsAll + linear: 0.11993551254272461  s
Accuracy of OnevsAll + linear: 1.0
Time for OnevsAll + polynomial: 0.3098316192626953  s
Accuracy of OnevsAll + polynomial: 1.0
Time for OnevsAll + gaussian: 0.20989251136779785  s
Accuracy of OnevsAll + gaussian: 0.9875
```

3) As expected, in general OnevsAll is faster than OnevsOne, since we have less comparisons to make. Accuracy for OnevsOne RBF is much higher than linear or polynomial, but naturally it takes more time.

4) In general, SVM is more accurate than Knn.

Part 3

Features I would consider are the number of people that own a car in different regions, to know which areas have the most people that are going to need a station. Also, the distance between each stations (so that we don't construct chargers in two gas stations that are very close for example, that would not be efficient). Another feature is how close to cities the gas stations are, the closer the more the demand increases since cities are more populated. We can also consider the number of drivers per day for various gas stations (drivers' throughput), the gas stations that attract the most drivers should be considered first. Another feature can be the range of a vehicle (how far can it go before the battery is dead), which would allow us to know where to place the chargers (if the car loses full battery after 1km of driving than maybe chargers shouldn't be placed in stations that are more than 1 km apart). Finally, we might consider the youth in our population in different areas. Younger people (20year olds to ≈40

year olds) will maybe tend to be more interested in electric cars than older generations. So the feature could be percentage of people that are between 20 and 40 years old in different areas.