Keshav Shankar

ECE 1395

Homework 5

**1b)**

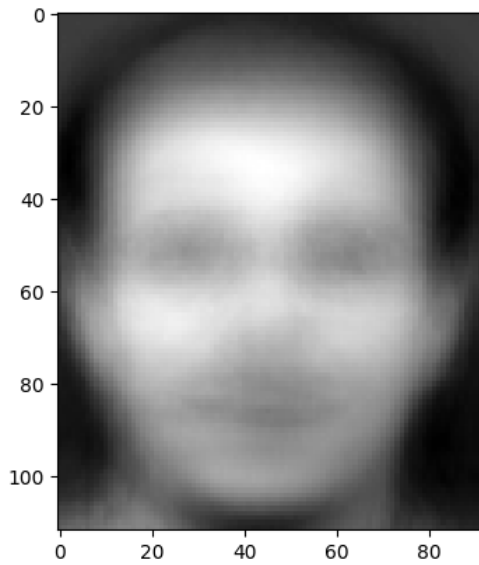| Sigma | Accuracy |
|-------|----------|
| 0.01  | 0.52     |
| 0.07  | 0.92     |
| 0.15  | 0.92     |
| 1.5   | 0.8      |
| 3     | 0.72     |
| 4.5   | 0.72     |

Sigma = 0.07 and 0.15 seemed to provide the best accuracy of 92%. Overall, increasing Sigma reduces accuracy.
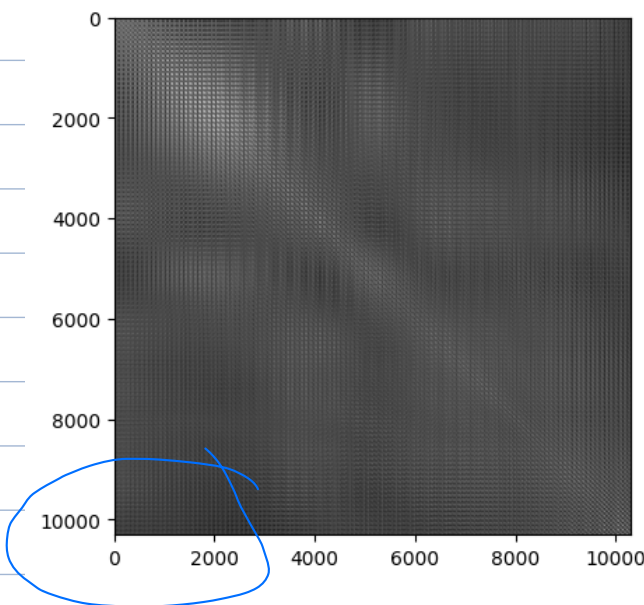
**2.1a)**
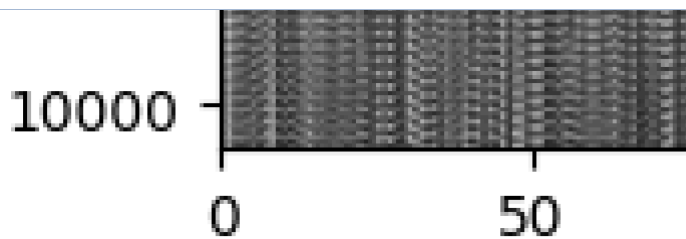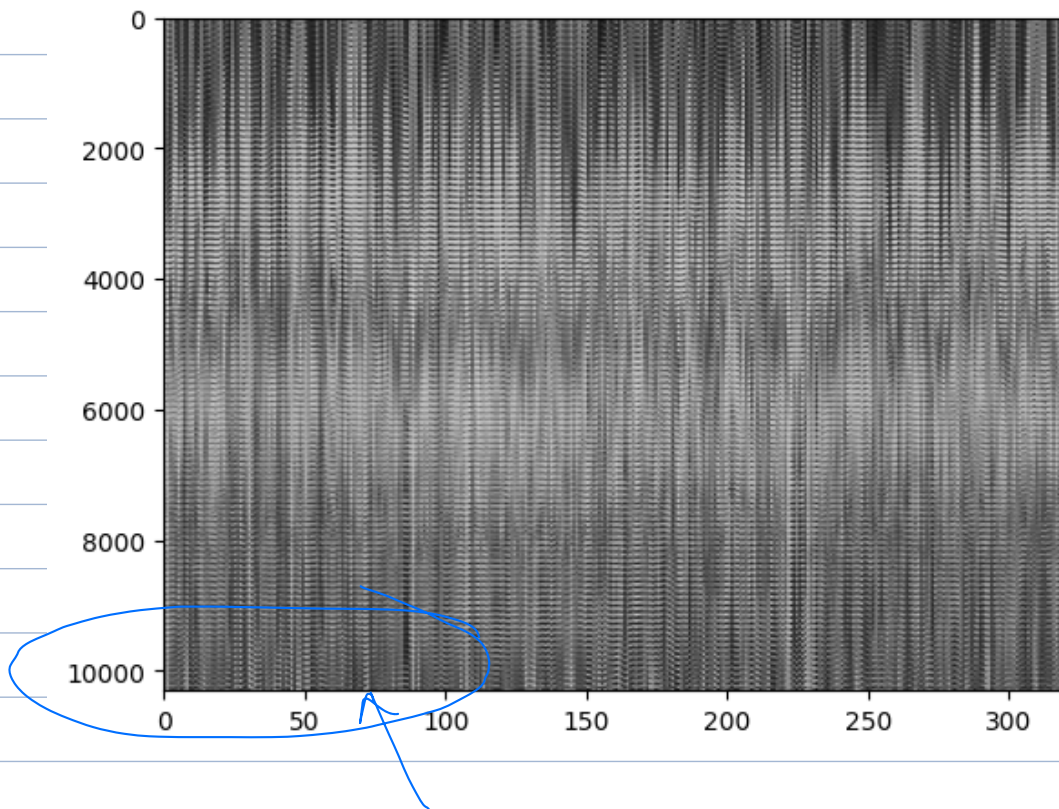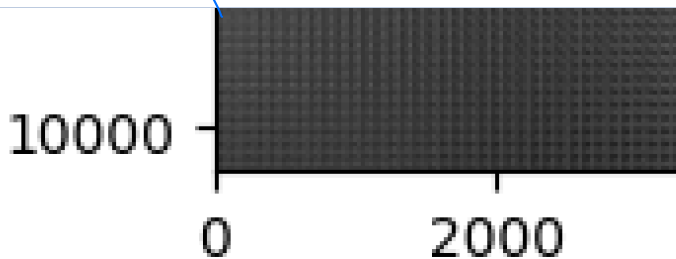


Zoomed in

## 2.1b)



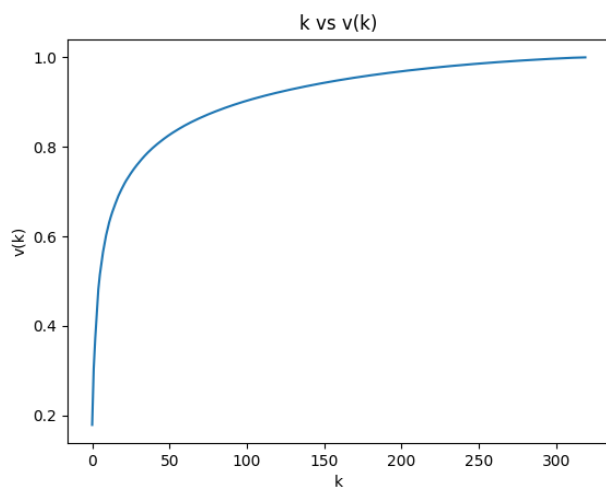Very blurry outline of a face. It does indeed look like all faces averaged together.

## 2.1c)



* I believe I used a dif. grayscale which is why this looks grayer than usual.

Zoomed in

**2.1d)**



k vs v(k)

K for 95%: 163

**2.1e)**



✳

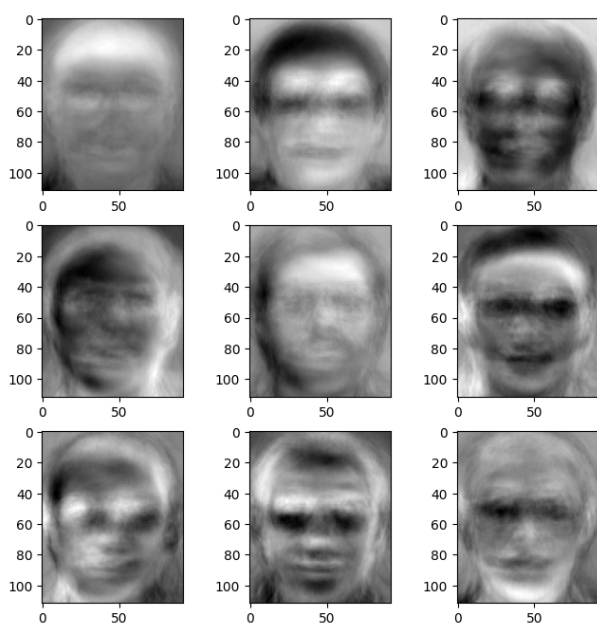Once again, looks a bit
more x-ray ish than normal
b/c of my grayscale

These do look correct though
as they resemble faces.

U has 10304 rows and 163 columns.

**2.2b)**

W_training has 320 rows and 163 columns.
W_testing has 80 rows and 163 columns.

**2.3a)**

K  1 | Accuracy  0.9625
K  3 | Accuracy  0.925
K  5 | Accuracy  0.875
K  7 | Accuracy  0.8
K  9 | Accuracy  0.7625
K  11 | Accuracy  0.75

**7.3b)**

```
    Kernel  Paradigm  Training Time (s)
0   Linear    OVR              0.018945
1   Linear    OVO              0.015368
2     Poly    OVR              0.017662
3     Poly    OVO              0.017589
4      RBF    OVR              0.022521
5      RBF    OVO              0.022375
    Kernel  Paradigm  Testing Time (s)
0   Linear    OVR              0.001881
1   Linear    OVO              0.001806
2     Poly    OVR              0.001845
3     Poly    OVO              0.001805
4      RBF    OVR              0.003820
5      RBF    OVO              0.003794
    Kernel  Paradigm  Accuracy
0   Linear    OVR      0.9750
1   Linear    OVO      0.9750
2     Poly    OVR      0.8125
3     Poly    OVO      0.8125
4      RBF    OVR      0.9625
5      RBF    OVO      0.9625
```

In general, one vs. one was faster than one vs. all
for every kernel; however, the accuracy on OVO is the same
as OVR paradigm.
  The accuracy of all SVM's outperformed KNN (1-11
because it handles outliers better.                    neighbors)
Within SVM, the linear kernel did best and took
  the least time to train and test, on average.


**3)** Traffic patterns — I want to know which routes/roads
are used the most across the country to place them on
those routes.

Population — I want to know which regions have the most
people, as they would need more chargers.

Future development — I want to know which regions have
best plans for development to better cater to their needs.

Environment - I want to know the weather patterns, as harsher conditions mean more maintenance for the chargers.

Vehicle Owners - I want to know what % of the population actually has cars, and of that, who has electric already.

Grid/Infrastructure - I want to know what regions actually have infrastructure to support the chargers.

In all, these should be the basics to make a good model to get the optimal locations for chargers.