Beryl Sin

PeopleSoft ID: 4522433

03/03/2024

Problem Set 5

1) a. See weightedKNN.py

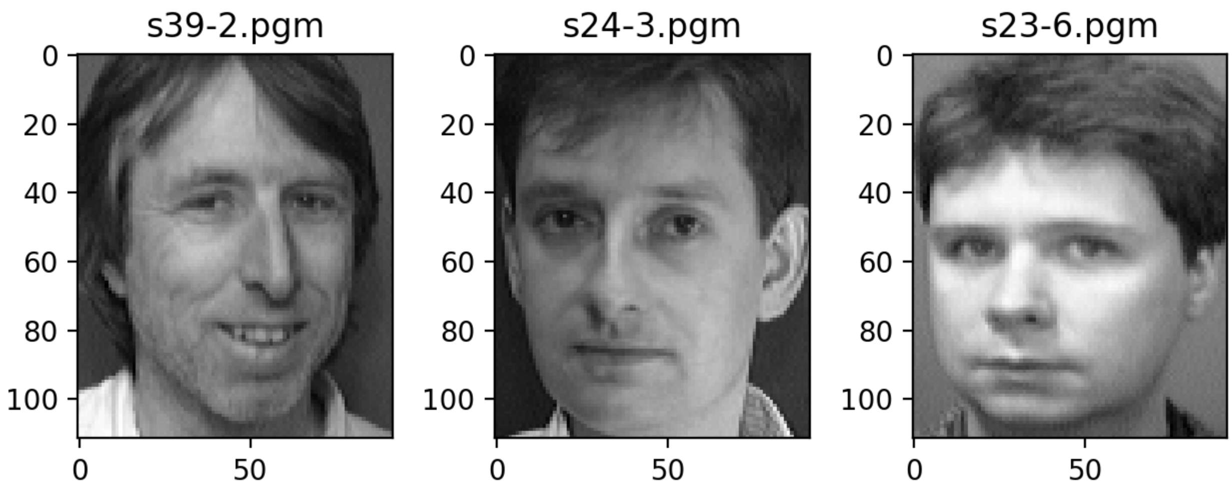
   b.

Question 1B :

Sigma	Training Accuracy	Testing Accuracy
0.01	1	0.68
0.07	1	0.92
0.15	1	0.92
1.5	0.912	0.8
3	0.88	0.72
4.5	0.904	0.72

Testing accuracy is highest between sigma values of 0.07 and 0.15. When value of sigma goes higher than 0.15, both training and testing accuracy start to decrease.

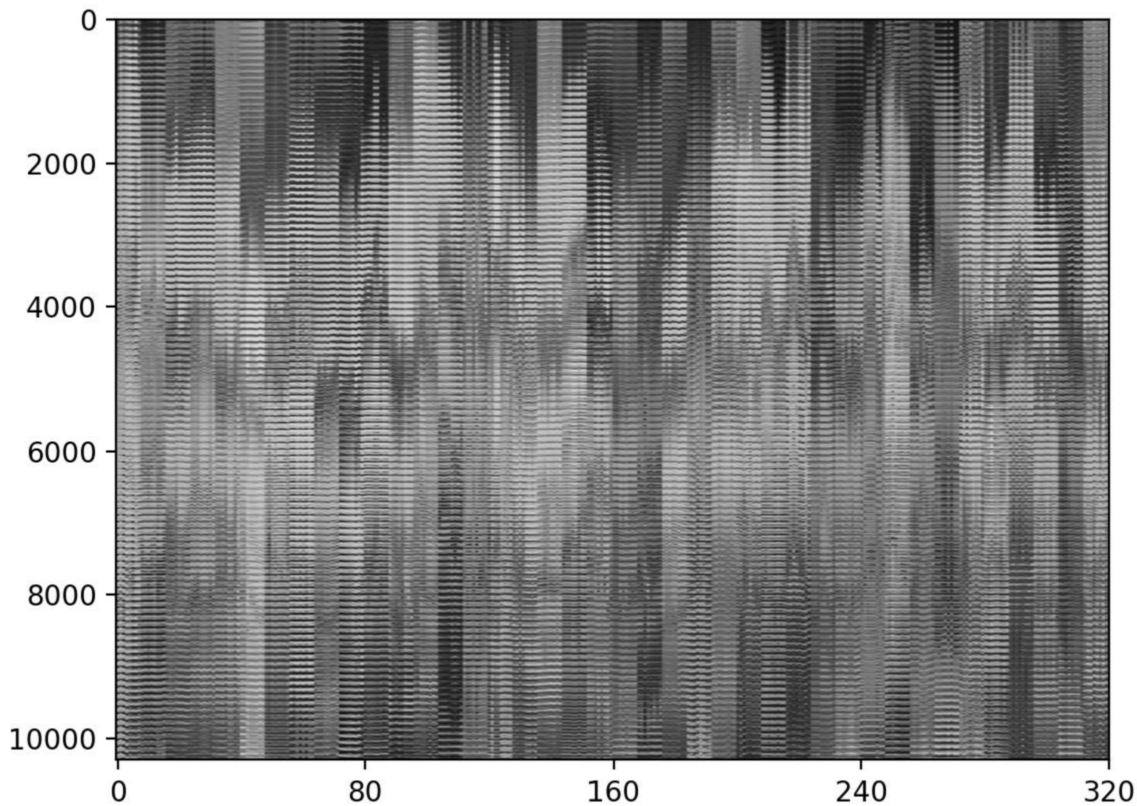
2-0) See ps5.py for file organization/code.



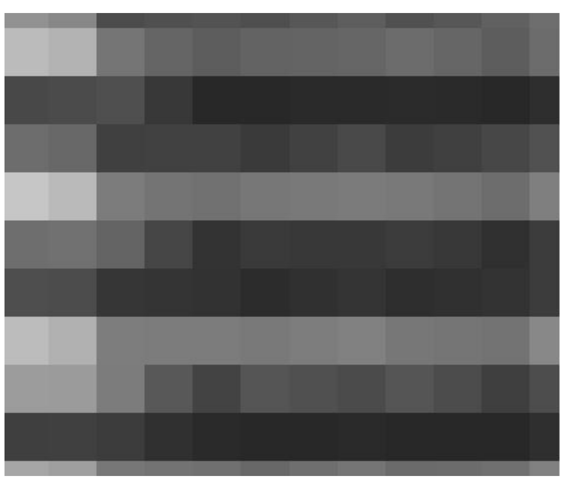
Images match what was read

2-1) a. See ps5.py for file organization/code

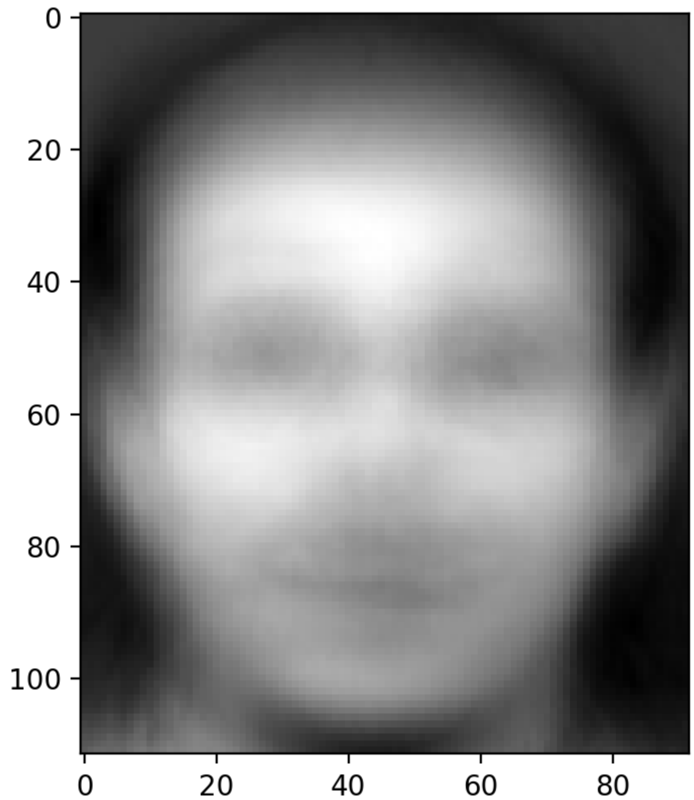
T (stretched the x-axis)



Random zoom-in:

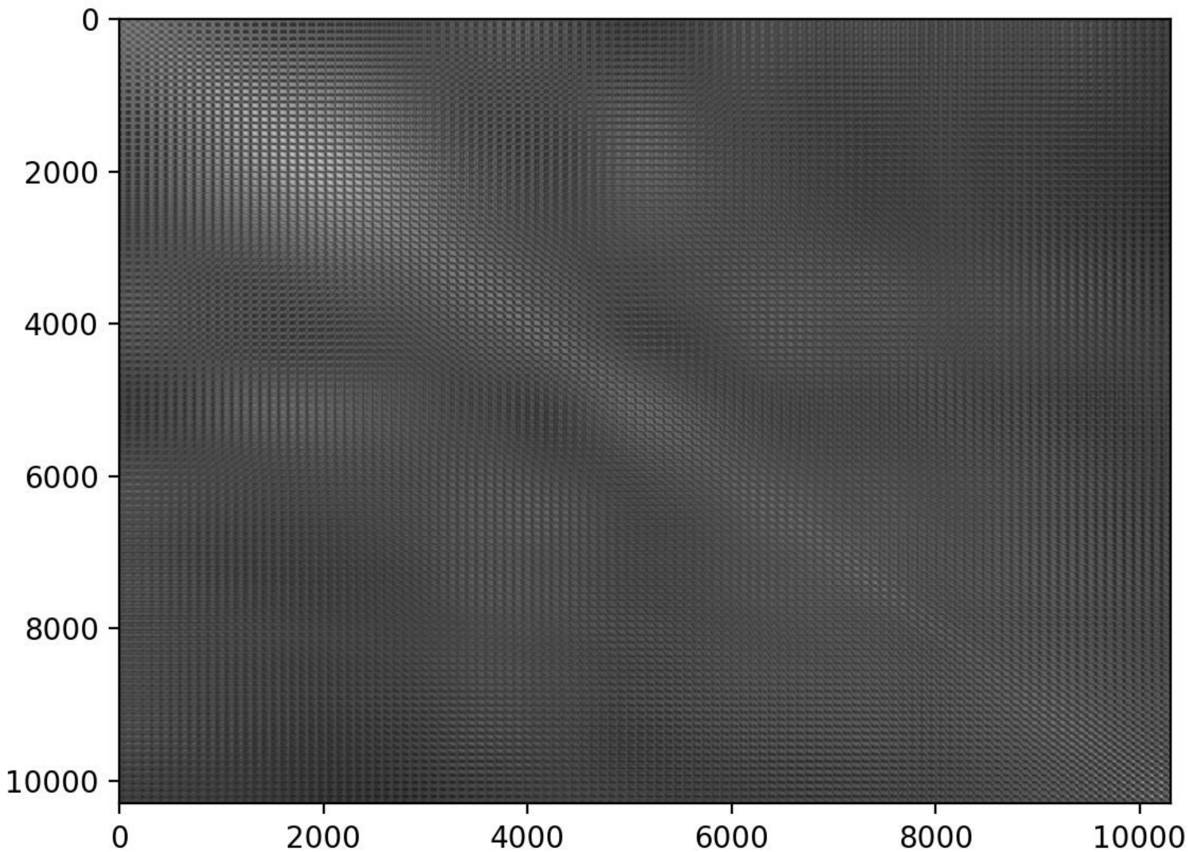


b. Mean face:

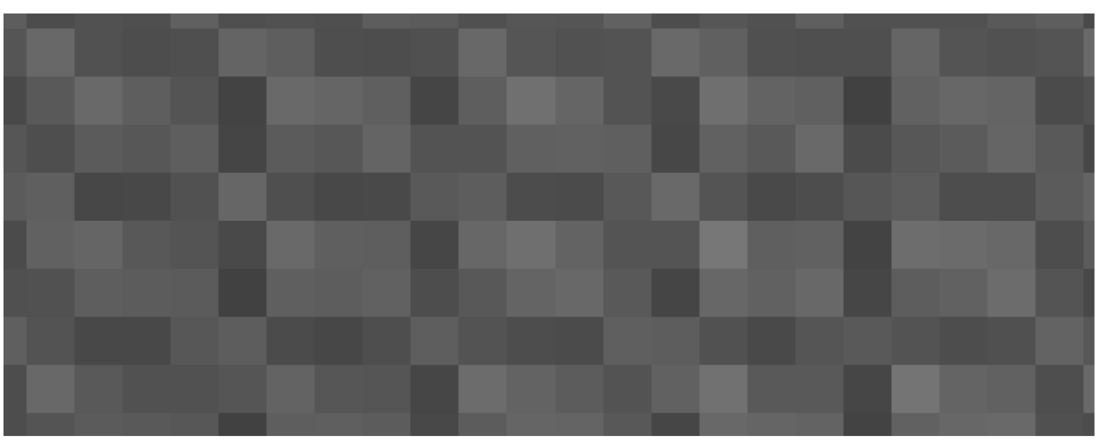


The mean face looks like a man’s face, but blurry.

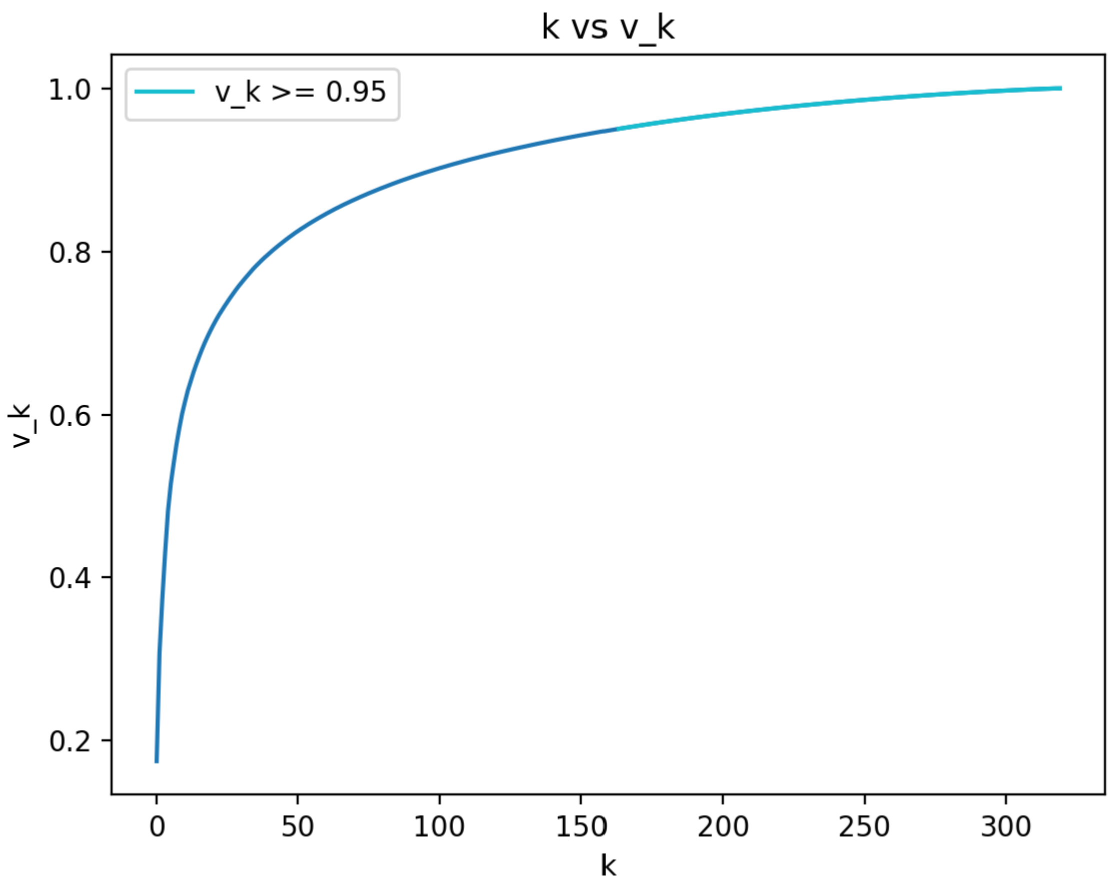
c. Covariance matrix:



Random zoom-in:



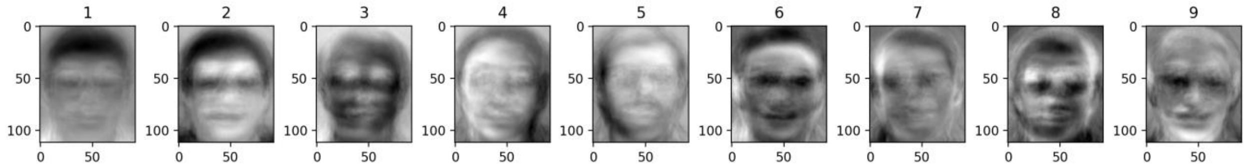
d. k vs v_k()




```
Question 2-1D, Number of eigenvectors: 162
```

Note: rerunning code may yield different number because of the randomness when selecting images from folders

e. See ps5.py for code



Faces look distinct and ghost-like, but they resemble faces

2-2) a. See ps5.py for code

b.

```
Question 2-2A, W_training.shape: (320, 162)
Question 2-2B, W_testing.shape: (80, 162)
```

2-3) a.

Question 2-3A :

K	Accuracy
1	0.0375
3	0.025
5	0.0125
7	0.0125
9	0.0125
11	0.0125

Testing accuracies are super low, but general trend: accuracy decreases as number of neighbors, K, increases b.

SVM Classifier	Training Time	Testing Accuracy
Linear (One-vs-One)	0.015625	0.0375
Linear (One-vs-All)	0.015625	0.0375
Polynomial (One-vs-One)	0.015625	0.05
Polynomial (One-vs-All)	0.03125	0.05
RBF (One-vs-One)	0	0.025
RBF (One-vs-All)	0.03125	0.025

Training time is in seconds.

Testing accuracies, again, are low. It seems like out of all the classifiers, the polynomial SVM is most accurate. Perhaps there is not enough data/too many classes, which impacts the accuracies, or there could have been some underlying issue when trying to project the eigenvectors onto the data samples. Using the One-vs_All method seems to take longer time than just doing the One-vs-One method. Additionally, it is clear that the polynomial SVM outperforms the KNN models.

3) Some features I would use for my model include (by city) the population density, cost of living, average salary, number of car owners, air quality, and gas station locations. I included:
   - Population density because generally the more people that live/interact with a city's highway, the wider the audience a charging station can service
   - Cost of living/average salary because that affects the kind of people that can buy/consider cars (as opposed to other modes of transportation)
   - Number of car owners, which correlate somewhat to population density and COL/avaerage salary, which is included because again, tells how many people in a city's highway that can potentially benefit from a charging station
   - Air quality because states with bad air quality may benefit from having accessible charging stations as a way to advertise this new technology
   - Gas station locations because people may already be familiar with where the gas stations are, so putting a charging station near them will make it more convenient for experienced drivers