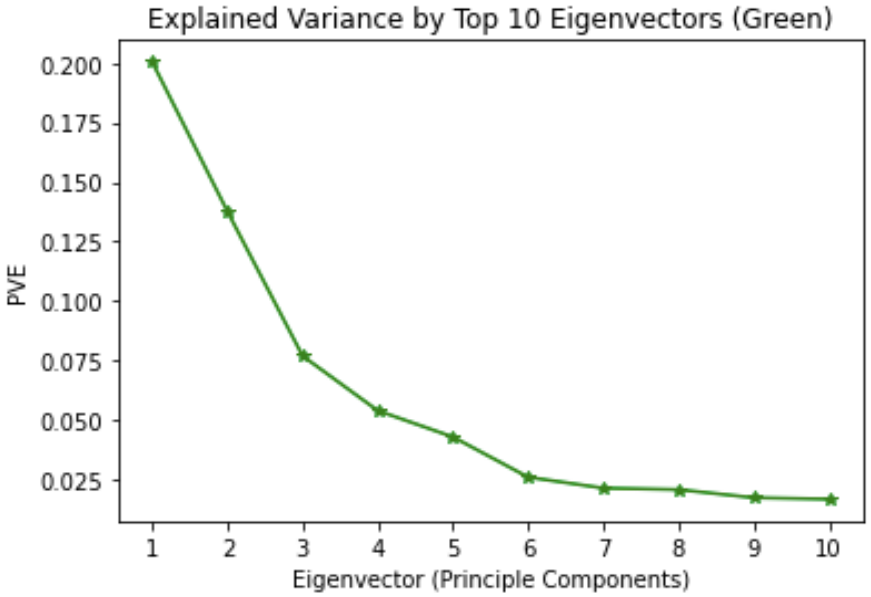
Necati Furkan Çolak
IE / 21803512
CS464-02

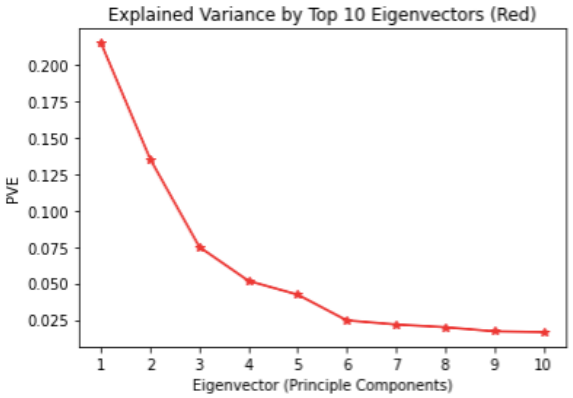
CS464-HW2 Report

Question 1.1:

PVE Results for green, red and blue can be seen respectively below.

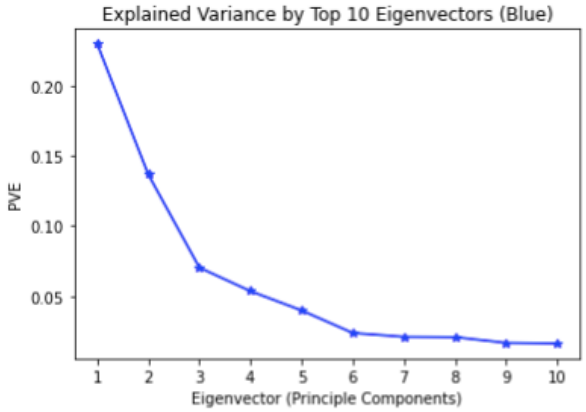



```
          PVE
1    0.200454
2    0.137676
3    0.076952
4    0.053970
5    0.042918
6    0.026022
7    0.021426
8    0.020812
9    0.017393
10   0.016811
Total PVE for top 10 eigenvalues:  0.6144337496251132
Required component amount for 70% variance: 19
```




```
          PVE
1   0.215068
2   0.135421
3   0.075041
4   0.051726
5   0.042286
6   0.024580
7   0.021770
8   0.019897
9   0.017070
10  0.016558
Total PVE for top 10 eigenvalues:  0.6194178151855934
Required component amount for 70% variance: 18
```

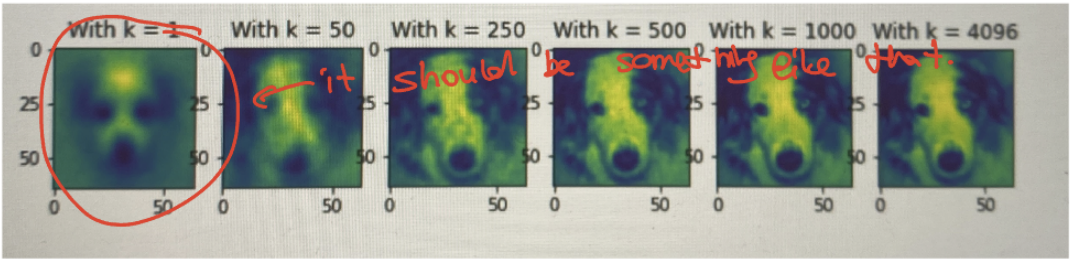
PCA for Blue

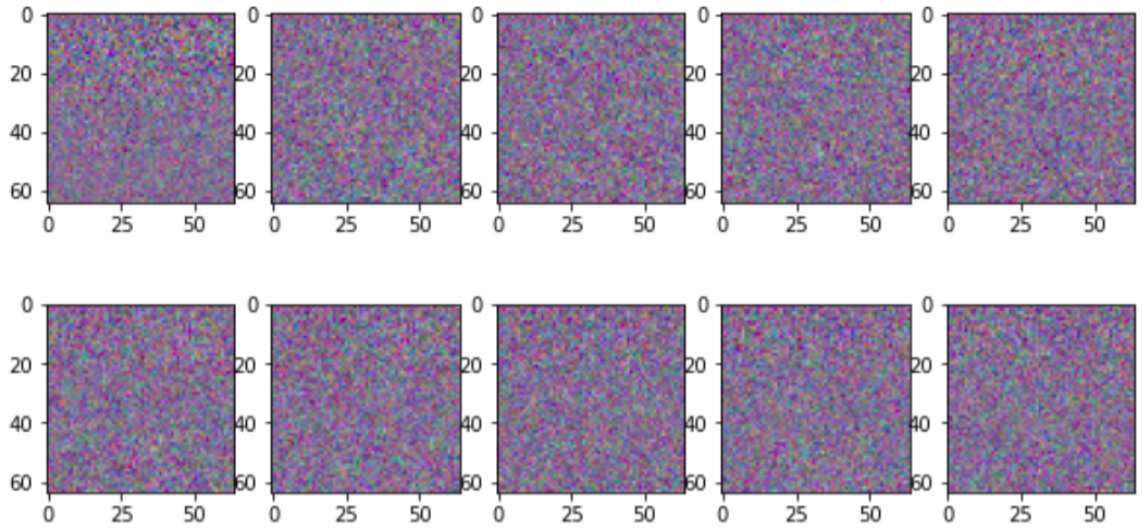



```
          PVE
1   0.229946
2   0.136770
3   0.070332
4   0.053559
5   0.039817
6   0.023731
7   0.020990
8   0.020757
9   0.016680
10  0.016291
Total PVE for top 10 eigenvalues:  0.6288728026196491
Required component amount for 70% variance: 17
```

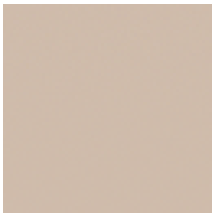
Question 1.2:

After the min-max scale for the top 10 principal components, the visuals can be seen below. However, I expect different results than these pixels. Since the results should contain some shapes that resembles dog barely. The reason why I expect this result is we take top 10 principal components and these principal components are the best ones to identify a dog since its variance high. However, these pixels do not provide anything useful for identification of dog pictures. When I try to solve this question, I got a result that resembles to answer. The result should resemble something in the related questions first image.

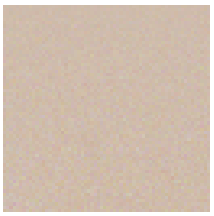




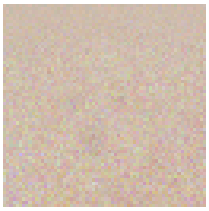
Question 1.3:



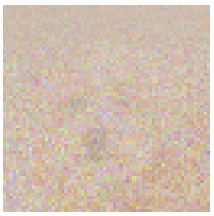
Result for k = 1



Result for k = 50

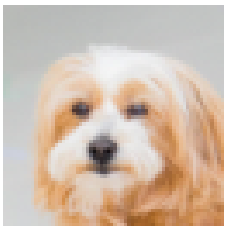


Result for k = 250



Result for k = 500

Result for k=1000



Result for k = 4096

To reconstruct an image, the dot product is applied to our centered feature matrix and our principal components. These calculations' results are used again and again as a dot product operant with the transpose of the principal components. After then, the mean vector was added to the result. The image was reshaped to a 64x64 matrix.

If we analyze these results, we can conclude that as number of principal components increases, the image becomes more precise.

Question 2.1:

- As our batch size decreases, run time takes much more time.

- In three methods, I get the highest accuracy from the Full batch. However, I expect that stochastic gradient ascent should have more accurate since features are taken into the model one by one. Then, In that case, my code has some computational mistakes.

Expected model performance: stochastic > mini-batch > full batch
My results: full batch > stochastic > mini-batch

Question 2.2:

- In this question, I get a better result in the normal distribution than in the uniform distribution.

Question 2.3:

- In this question, different learning rates are used. The best learning rate among all possible learning rates is 10^{-3}

- As learning rate increase, the speed of algorithm decreases.