CS-464
Introduction to Machine Learning
Homework 2 Report

Kaan Tek
21901946

Question 1.1

In order to obtain the principal components, I used the *numpy.linalg.eig* function, which returns both the eigenvalues and eigenvectors (principal components). Then, I computed the PVE for each principal component by dividing the corresponding eigenvalue to the sum of all eigenvalues. The results are below:

Red Channel:

```
indexes of the largest eigenvalues:  [0 1 2 3 4 5 6 7 8 9]
PVE for each principal component:
 [0.23505564 0.15650303 0.09004832 0.06829707 0.03752532 0.02394648
 0.02276349 0.02112766 0.01793531 0.01349271]
total variance explained by top 10 principal components:
0.6866950290522249
min number of principal components needed for >= %70 PVE:  12
```

Green Channel:

```
indexes of the largest eigenvalues:  [0 1 2 3 4 5 6 7 8 9]
PVE for each principal component:
 [0.2087307  0.15883864 0.09258536 0.06810819 0.03798398 0.02446589
 0.02427818 0.02148947 0.01886943 0.01421092]
total variance explained by top 10 principal components:
0.6695607576002373
min number of principal components needed for >= %70 PVE:  13
```

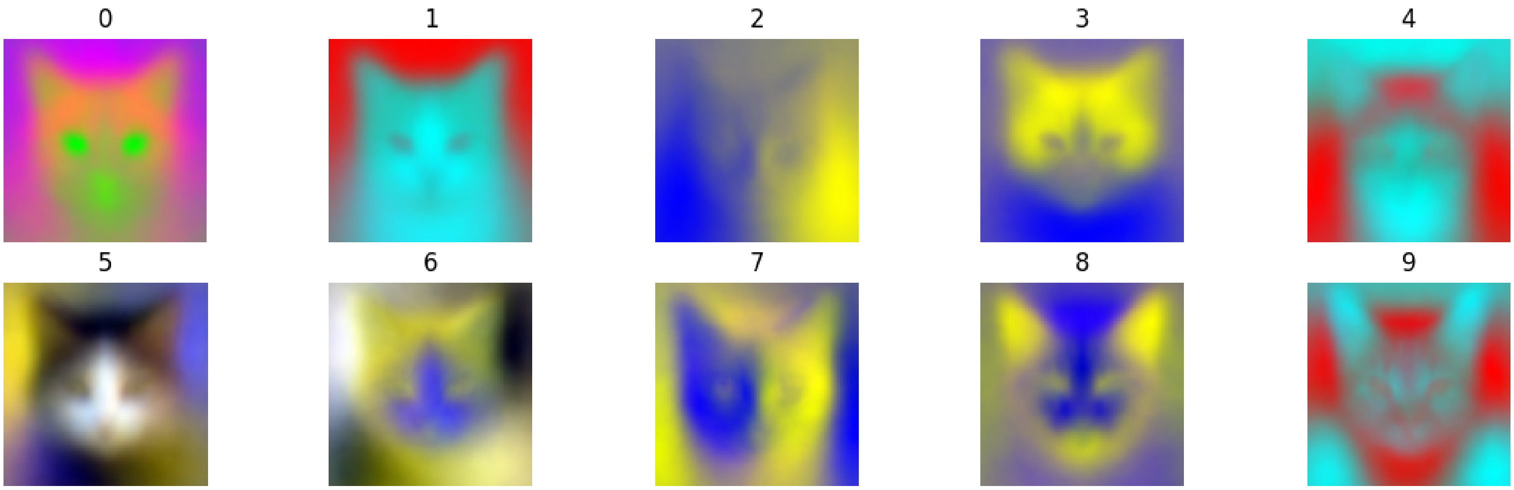
Blue Channel:

```
indexes of the largest eigenvalues:  [0 1 2 3 4 5 6 7 8 9]
PVE for each principal component:
 [0.22855928 0.15647814 0.0878948  0.06202809 0.0373974  0.02416357
 0.02404514 0.0205939  0.0184576  0.01428438]
total variance explained by top 10 principal components:
0.6739023171575512
min number of principal components needed for >= %70 PVE:  13
```

It can be seen from the above results that for each color channel, the first 10 principal components can capture slightly less than %70 of the total variance. One important mark is that for all three color channels, the first two principal components capture a great amount of variance, which is more than %35 for both of them in total. In order to capture more than %70 of the variance, at least 12, 13, and 13 principal components are needed for the red, green, and blue channels, respectively.

Question 1.2

In order to obtain visuals of first 10 eigenvectors, I stacked i^{th} principle component of each color channel on top another where i takes values from 1 to 10. It can be said that these images are a great representation of all original images since these are the top 10 principal components and they capture most of the variance, almost % 70. By linearly combining these images, one can obtain a close approximation of the original images. Below are the resulting ten images:

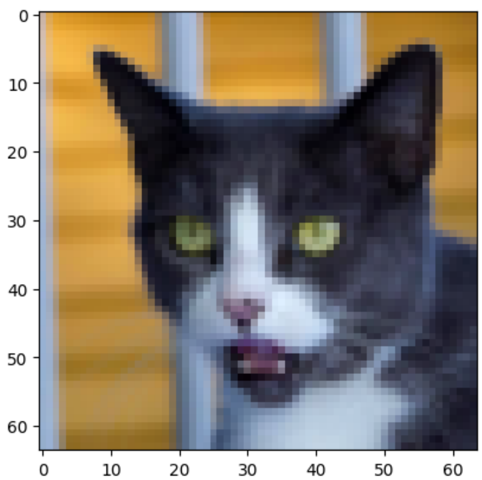


Question 1.3

The first step in this question was to apply dimensionality reduction to the original image, in other words, projecting the image on its first k principal components. In order to achieve this, we first subtract the mean from the image to center the data, and then take its dot product with the transpose of the first k principal components, in order to obtain a k-dimensional vector as the output.

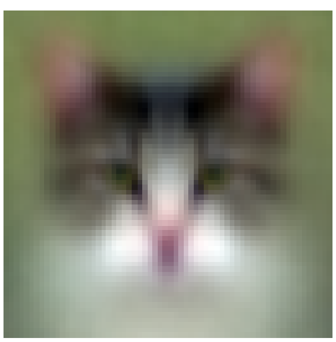
The second step is to transform the data back to its original space, in other words to reconstruct the original image by using first k principal components. This is done by taking the dot product of the output matrix of the first step with the first k principal components, and then re-adding the mean. This operation gives us a matrix with dimensions (4096, 3), which is the same as the original image. Note that these steps are done for each color channel and the results are stacked together. The results are below:

Original image:

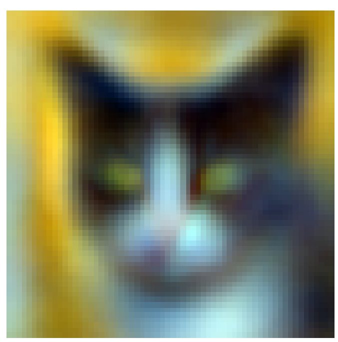


Reconstructed images:

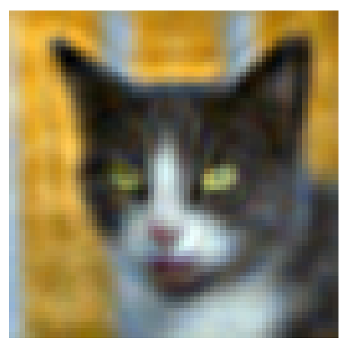
Reconstruction with k=1



Reconstruction with k=50



Reconstruction with k=250



Reconstruction with k=500

Reconstruction with k=1000

Reconstruction with k=4096

As can be seen from the above results, using more principal components for the reconstruction gives a better approximation of the original image. This makes sense because using more principal components indicates taking into account more variation. It can also be noted that even by using only 1 principal component, it is predictable that the original image is related to a cat. Also by using ~100 principal components, one may be able to predict the exact original image from the initial dataset. Finally, using 4096 principal components gives almost if not the same image as the original one, since all principal components are being used for reconstruction.