Bilkent University

CS-464

Spring 2022/2023

Homework-2

02/05/2023

Name	Student ID	Section
Halil Alperen Gözeten	21902464	01

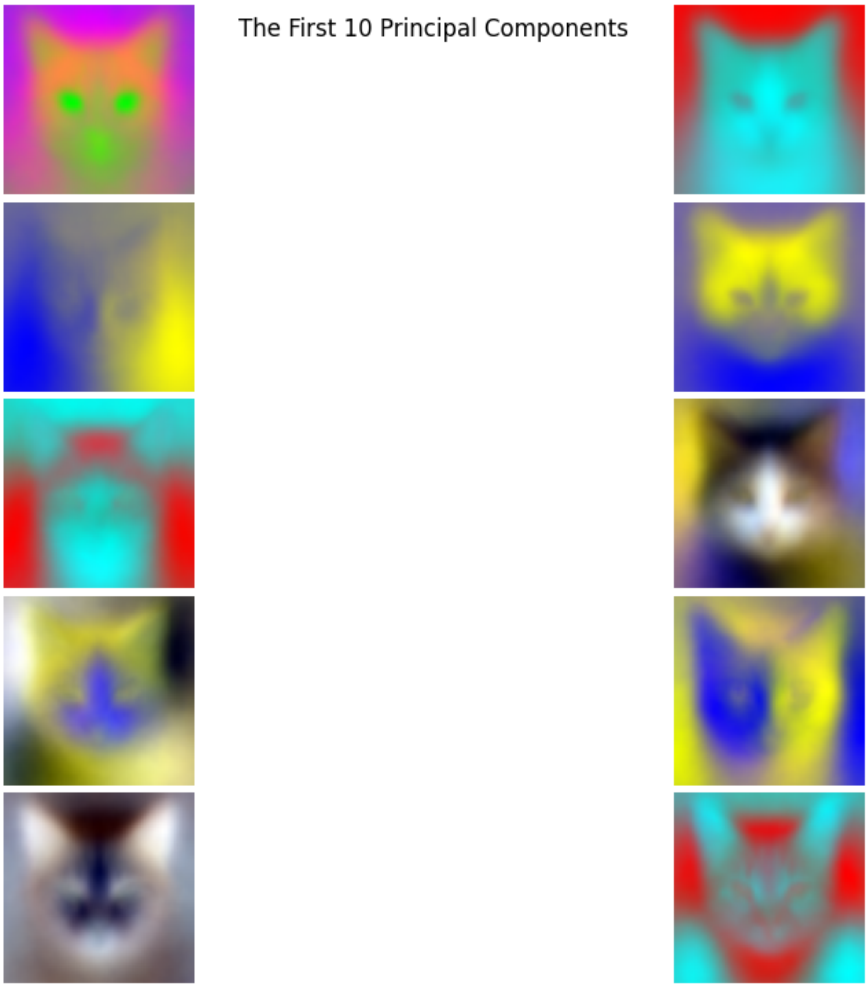
Question 1.1

In order to find the explained variance ratios, one can divide the sum of the selected corresponding eigenvalues to sum of all eigenvalues. After implementing the code in this direction and when the code for Question 1.1 is executed, the following output is obtained, which contains information about all channels:

```
------- Red Channel -------
Individual explained variances of first 10 PCs:
[0.2350697  0.15651115 0.09005254 0.06829955 0.03752734 0.02394754
 0.02276466 0.02112821 0.01793592 0.01349361]
Total explained variance by first 10 components:
0.6867302133538649
The minimum number of principal components to ensure %70 explained
variance: 12
------- Green Channel -------
Individual explained variances of first 10 PCs:
[0.20873715 0.15884566 0.09258857 0.06811112 0.03798505 0.02446732
 0.02427916 0.02149053 0.01887    0.01421134]
Total explained variance by first 10 components:
0.6695858942373207
The minimum number of principal components to ensure %70 explained
variance: 13
------- Blue Channel -------
Individual explained variances of first 10 PCs:
[0.22859036 0.15649258 0.08790596 0.06203548 0.03740134 0.02416587
 0.02404733 0.02059613 0.01845899 0.01428572]
Total explained variance by first 10 components:
0.6739797747155609
The minimum number of principal components to ensure %70 explained
variance: 13
```

It is observed that the total explained variance by the first 10 principal components for each channel is less than 70% (around %67), which means that we need more principal components to capture a significant proportion of the variance in each channel. Also, from the above outputs, the minimum number of first principal components required to obtain %70 explained variance is quite similar for red, green, and blue channels, which are 12, 13, and 13. Here, we also see that more than %20 of the variance is explained by the first principal component and more than %15 by the second principal component for all three channels.

Question 1.2



The above images are obtained by stacking the first 10 principal components for red, green, and blue channels so that the i^{th} principal components of colors are stacked together. Most of the variance (around %68.6, %69.5, %67.3 for red, green, and blue channels, respectively) for a cat image comes from the above components. To put up differently, these images carry the majority of the information contained in the original cat images, and a good approximation of the original images can be obtained using the linear combinations of these images.

Question 1.3

In this part, the first cat image, 'flickr_cat_000002.jpg' has been selected for demonstration, and it has been reconstructed using k ∈ {1, 50, 250, 500, 1000, 4096} principal components.

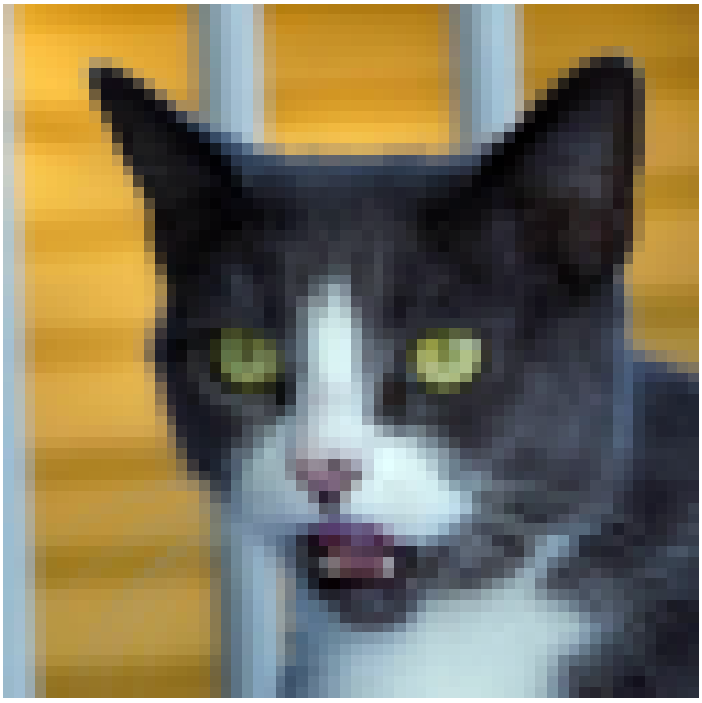
The mean subtracted (centered) data is first projected by taking its dot product with the matrix V, which is formed by selecting the first k principal components, and the result is a k-dimensional vector. Then, the inversion is done by calculating the dot product of the resulting vector with the matrix V, and the mean is added back to this resulting 4096-dimensional vector. Since the eigenvectors are orthonormal, we can use the transpose of the matrix V to reconstruct an image from its principal components as it's equal to the inverse of the matrix V; thus, simply taking the dot product with matrix V is enough. Then, the final obtained vector is reshaped to 64x64 and stacked for three different channels to obtain an image with dimensions (64,64,3) which has been then plotted using min-max scaling on each channel.

From the below images, we see that as k which is the number of first principal components used in the reconstruction increases, the quality of the reconstructed image also increases, details becomes more accurate, and the image becomes more similar to the original one. One important thing is that even a single principal component is enough to capture the cat face, as can be seen from below image in k=1 case. Moreover, even for 50 principal components, the reconstructed image mostly captures all significant details of the image. Also, for k=4096, there's a minor or no difference between the original and reconstructed image as the output for the below code is True.

```
    # Check if original and reconstructed images are identical within a tolerance
    if k == 4096:
        print(np.allclose(reconstructed_image, orig_image))
```

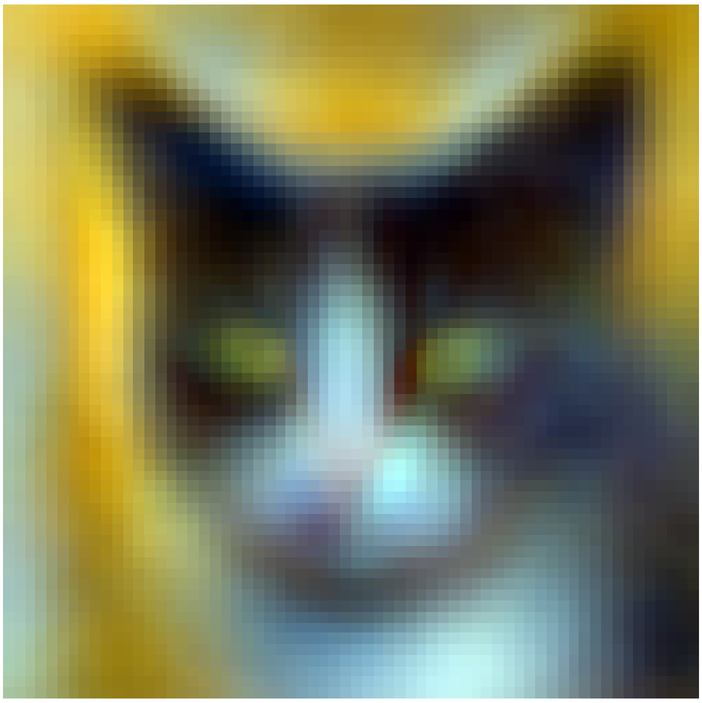
Output: `True`

Original Image



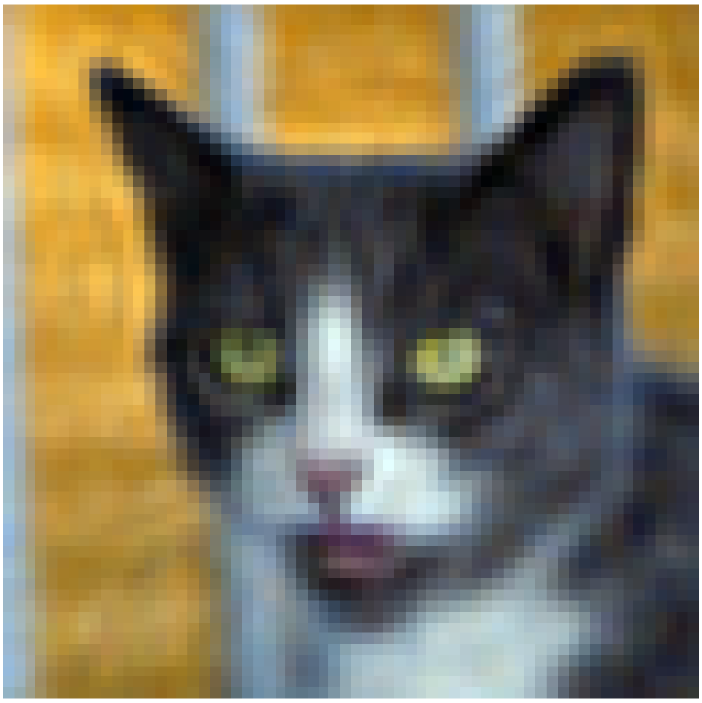
Reconstruction With k = 1 PCs

Reconstruction With k = 50 PCs

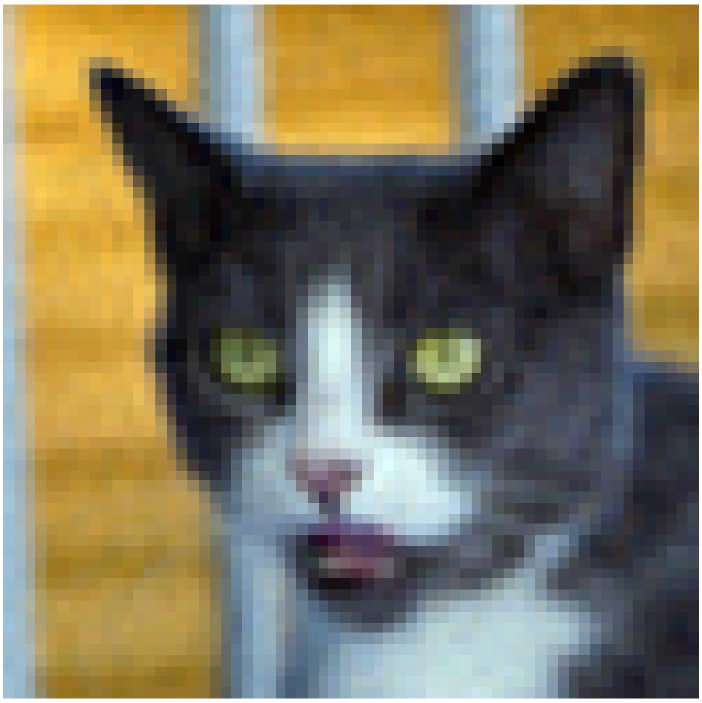


Reconstruction With k = 250 PCs

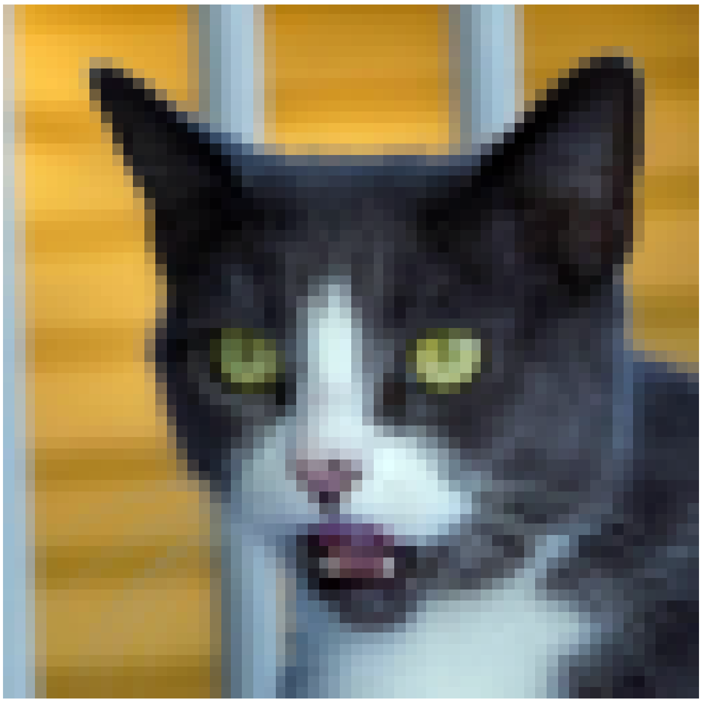
Reconstruction With k = 500 PCs



Reconstruction With k = 1000 PCs



Reconstruction With k = 4096 PCs



Original Image