

**CS-464 INTRODUCTION TO MACHINE LEARNING**

**HOMEWORK-2**

Doruk Barkın Durak

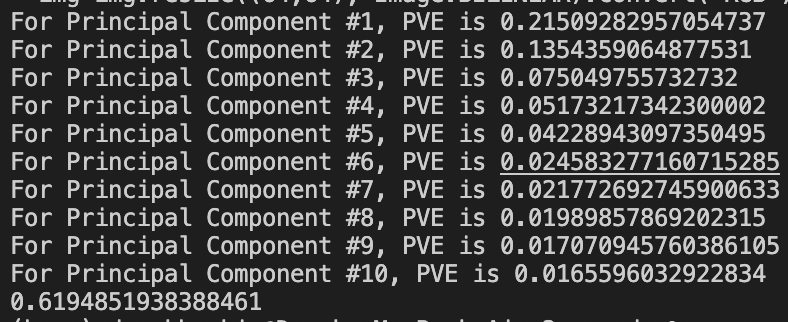
21802255

Section-2

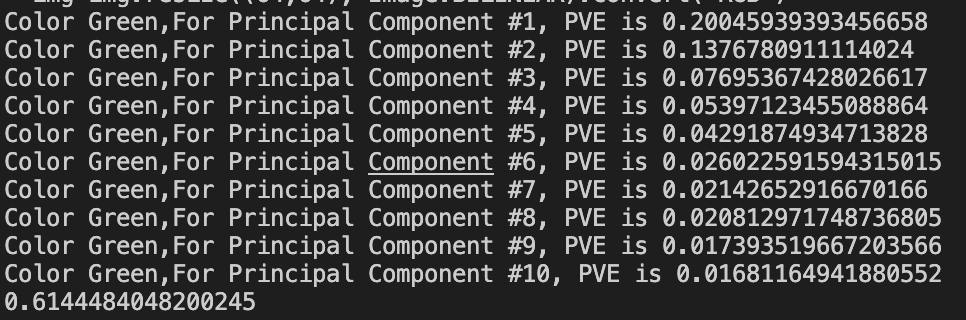
**Question 1:**

To apply Principal Component Analysis to data, first I mean centered the data for each different color and compute the covariance matrices for each. After, with respect to covariance matrix I calculated the eigenvalues and eigenvectors of each color. I sorted those eigenvalues with respect to principal components to find the necessary number of principal components.

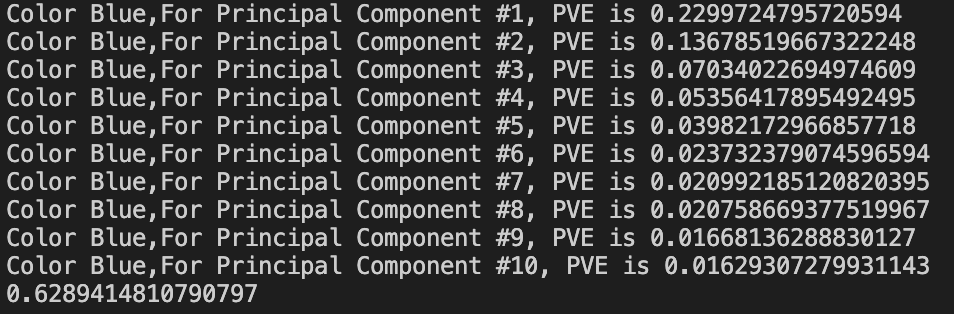
**1.1:**

Explained proportion of variance for first 10 principal components for color red:

Explained proportion of variance for first 10 principal components for color green:

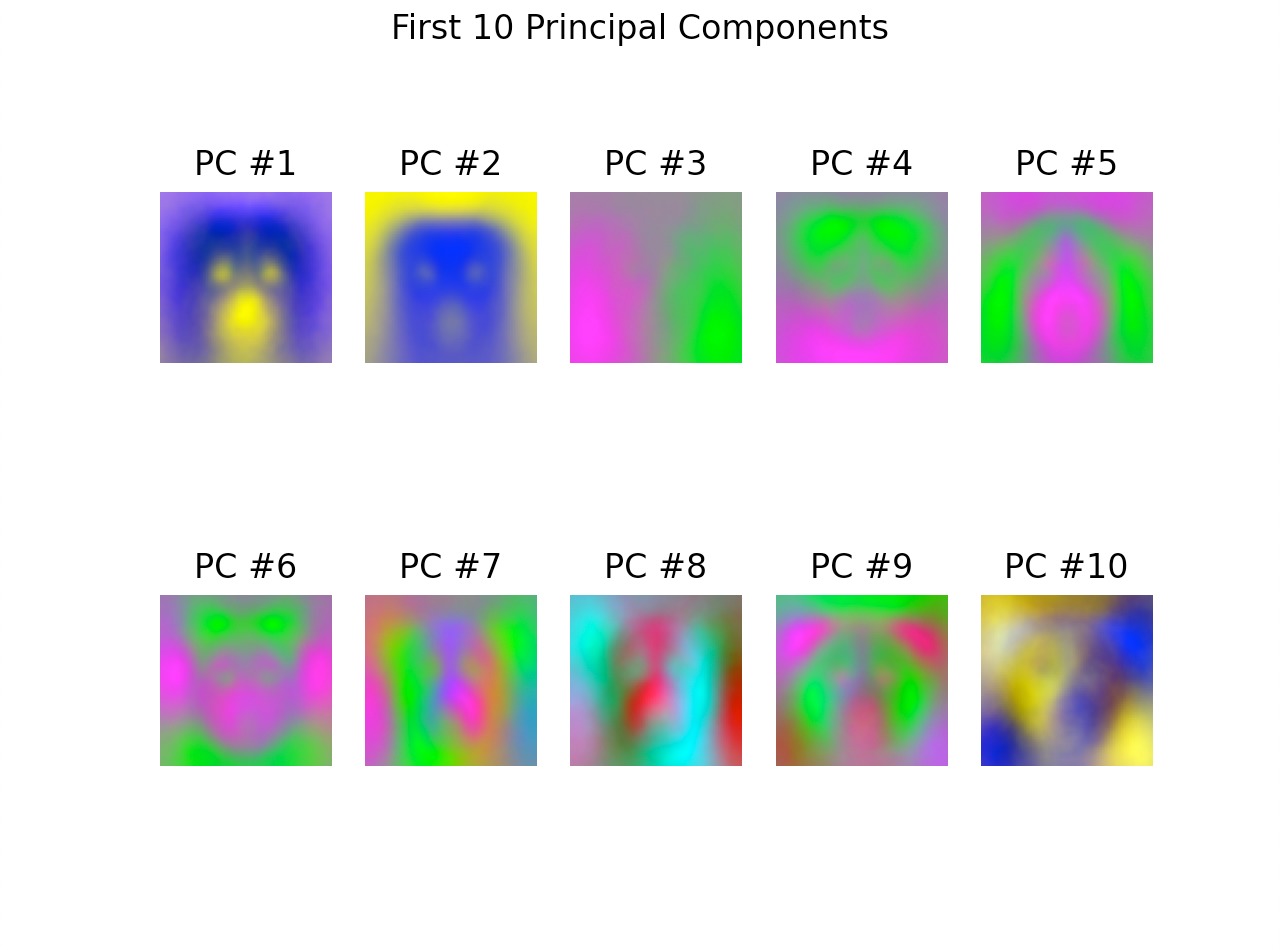


Explained proportion of variance for first 10 principal components for color blue:



As seen in the results. First principal component can explain approximately %22 variance for color red, %20 for color green and %23 for color blue. First 10 principal components are roughly estimating between %62-63 of variance in given data. Since we have a grand total of 4096 principal components for each color type, we can say that first 10 principal components are significant for explaining the variance. In my calculations for approximately %70 variance capture, we need first 17 principal components of color blue, first 19 principal components of color red and green. For variance capture, blue seems to be the best choice when we look at the statistics

**1.2:**



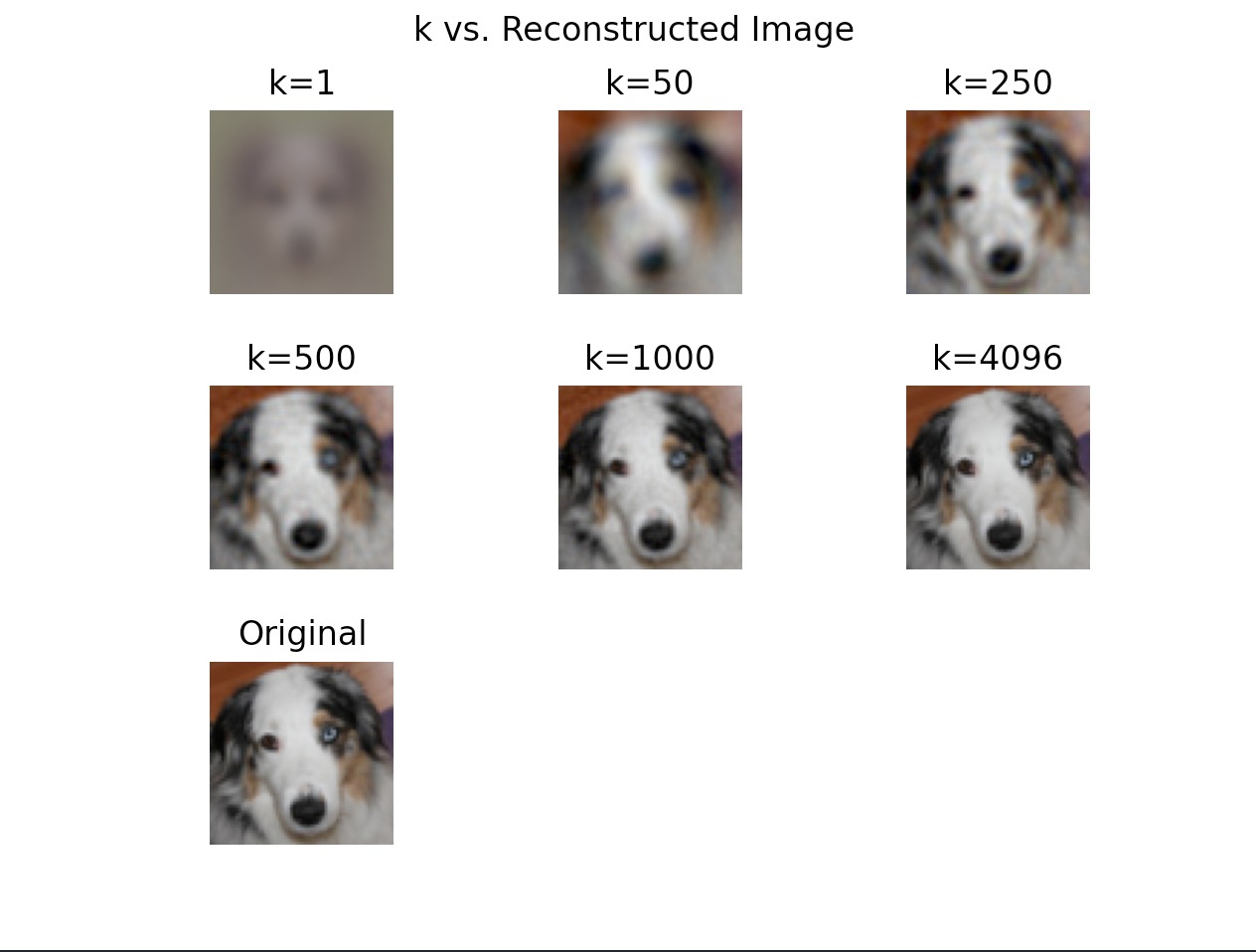
1st, 2nd ,9th, and 10th principal components can be recognized as dogs, but other principal components are tough to identify.

**1.3:**

A data matrix of A with m samples x n features. To find first k principal components, we need to center the dataset by subtracting mean of the dataset from every sample point. Let mean centered data be X and let another matrix V be the matrix that has first k eigenvectors. Columns of V are eigenvectors and V is a n x k matrix. So, by multiplying X x V we create a m x k matrix where this matrix includes Principal Component Analysis scores. To reconstruct we need to make this matrix again an m x n matrix, so we need to make k to n. If we take the transpose of V matrix and multiply it with above multiplication and add the means back. We will find:

X\_RC=X x V x VT + Mean

In our case we have 3 separate matrices, each Xi matrices will contain 5239 samples and 4096 features. V will be a matrix of 4096xk where k= [1,50,250,500,1000,4096]. Reconstructed dataset will be 5239x4096x3 because since we have 3 separate color categories, to reconstruct the images, we will need all 3 colors.



At k=1, we can identify it as a dog, but it is too blurry to make any kind of comments. At k = 50 some color details start to show up and we can now say that this is a 3D image of a dog because we can hardly identify the nose. At k=250, we can see more characteristics of the dog such as color pattern, ears, different eye colors. At k=500 details little bit clearer such as overall shape of the mouth and jaw. Since the component numbers increase, we have access to a wider color pattern, it is not surprising. At k= 1000 image is clear, only hard to see part is the iris of the dog. By comparing k=1000 to k = 4096 although image still improves but k = 1000 we have all the necessary details we have. It is safe to say that after 1000 principal components, increase in number of components is not effective. To prove this claim. I made to plots which is available at here: