Question A

Part 1

Principal Component of 3600 image patches of size 16x16 were calculated and then were reconstructed and combined to form reconstructed image.

This was done for K = 1, 10, 100, 150, 200 and 500.

The reconstructed images were stored for K =10 and K=100 with jpg format. These images can be found in Images -> Output Images -> A_1 folder in the repository.

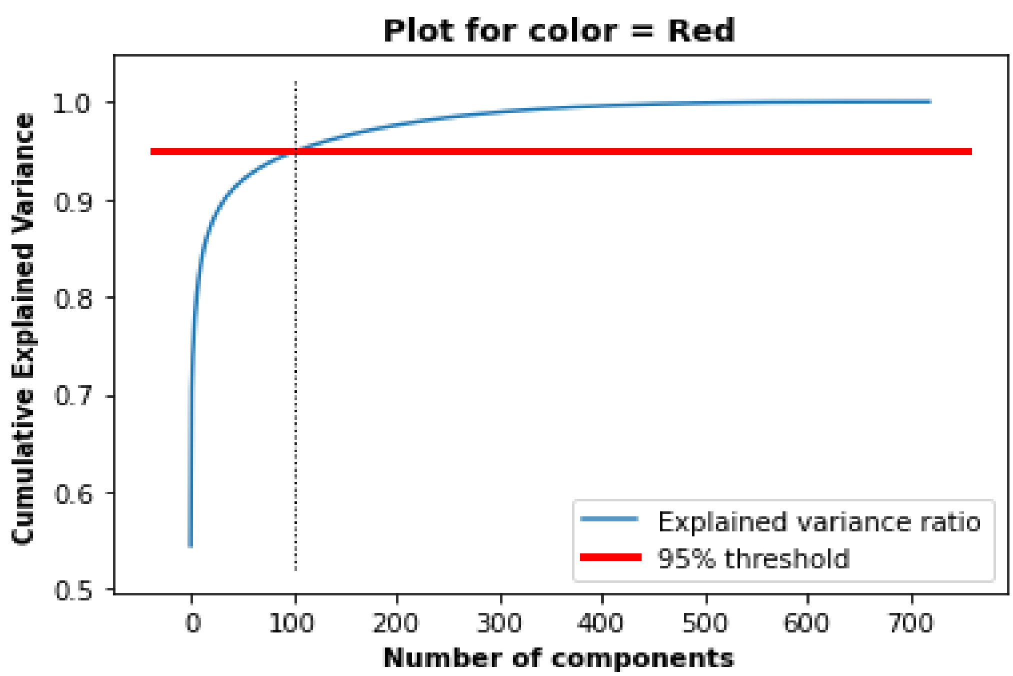
Part 2

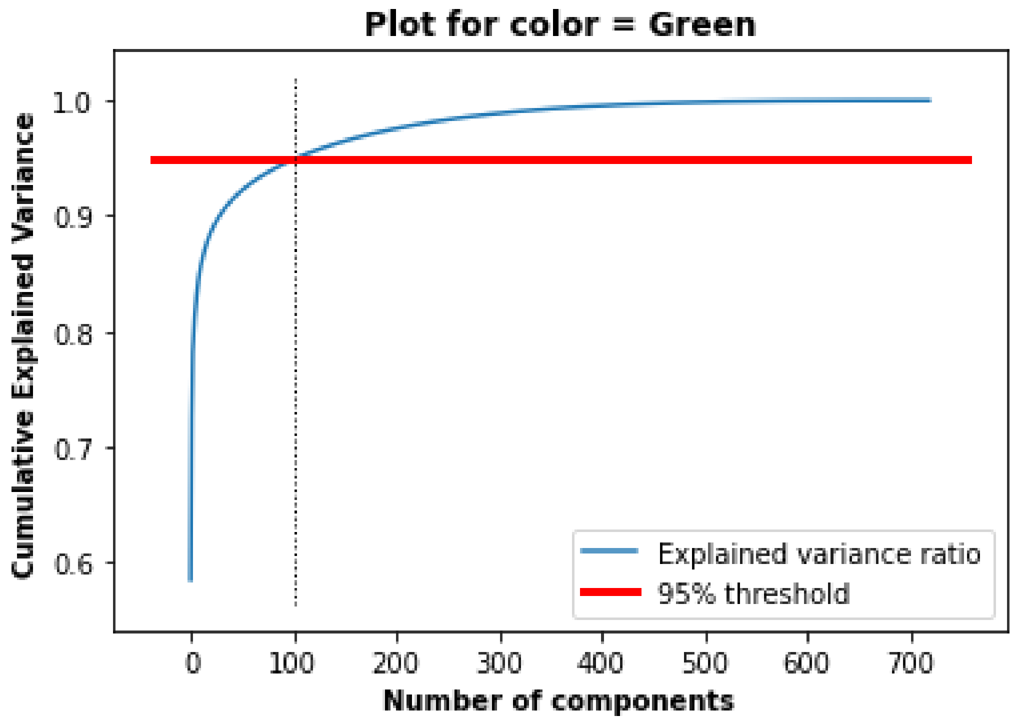
Top 10 highest principal components were evaluated using sklearn and then were displayed and stored as 16x16 images in the folder Images -> Output Images -> A_2 folder in the repository.

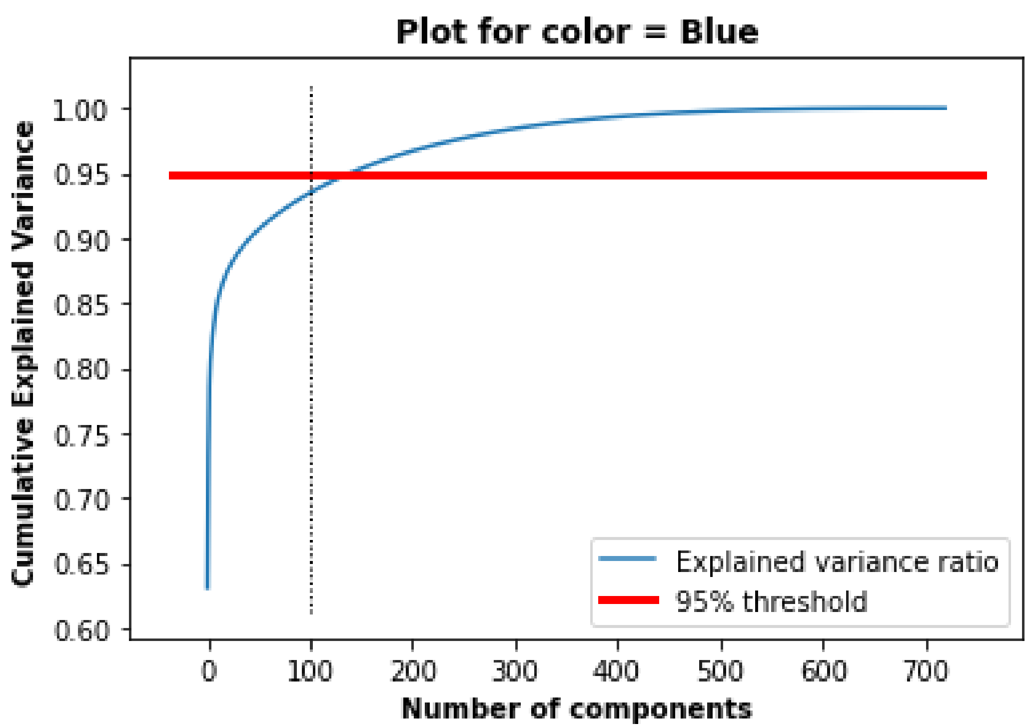
Part 3

Principal components are those eigenvectors what have the highest eigenvalues. Higher eigenvalues means higher density in the image, and so eigenvectors points towards maximum density of the image. If we more towards the direction of highest eigenvector, we will reach the point with maximum density. Whereas all the other points can be ignored, which do not contribute towards the image density as much as eigenvectors with highest eigenvalues does. This is principal component analysis.

So, we want to choose K (total number of principal components) in such a way for which the variance(density) is maximum for eigen values. I plotter the cumulative variance wrt the number of PCA components and got the following results.







As we can see, with increasing value of K, the variance increases as well, which is obvious since we are taking more and more components from the original image. However, taking into consideration the 95% threshold which is good enough for PCA,

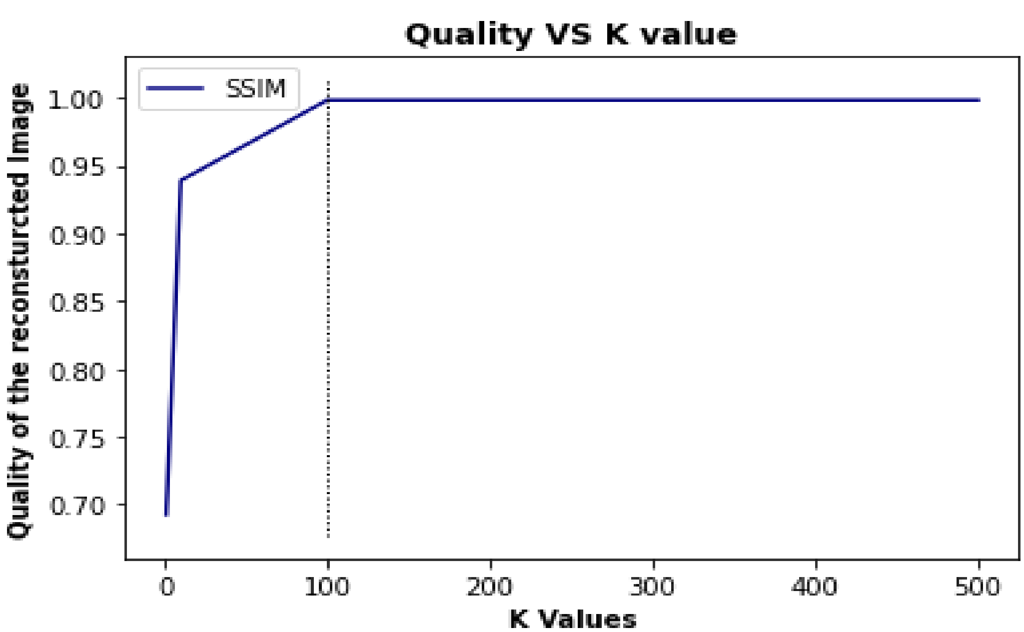
- K = 100 is best choice for Red
- K = 100 is best choice for Green
- K ~ 120 seems to be the best choice for blue.

All things considered, I believed K = 100 is the best choice for K value. In this part, the reconstructed image for K = 100 is not stored as it has already been done in part 1. The reconstructed image looks like –

Part 4

In this part, the quality of the reconstructed images for K = 1, 10, 100, 150, 200 and 500 is evaluated.

For quality assessment, Structural SIMilarity (SSIM) is used. It computes the similarity between two images. In my code, I calculated the similarity between the reconstructed images and original images. Then I plotted the SSIM value w.r.t. the K values. The plotted graph looks like –



As we can see, the similarity between the reconstructed images and the original images increases with K. This is obvious t understand as higher the value of K, more principal components will be used for reconstruction and more the reconstructed image will look like the original image.

As we can see from the plot, the similarity index remains the same after K = 100. So, there is no need to take any value of K greater than 100 as the results don't change and only the computational time and spaces increases.

This also validates the results of Part 3 for choosing 100 as the best value of K.

Question B

In Question B, I choose Image stitching project for this mid-term. Images for stitching are downloading from - `towardsdatascience.com/image-stitching-using-opencv-817779c86a83,` @credits – Vegdevi Kommineni.

For this project, openCV version – 3.3.0.10 and python configuration 3.3.0.10 is used. This is because SIFT() isn't a free function for openCV function greater than 3.4.4.44.

The main key points of this project is –

1. Import Images
2. Convert into gray scale, for similarities in colored photos, similarities in gray-scale images can also be used. SIFT() function isn't compatible for 3D numpy arrays.
3. Key points and sift descriptors are calculated for both of the images.
4. Descriptors are matches using brute force matcher – BFMatcher, using knnMatcher which matches the descriptors based on the Euclidian distance between two descriptors.
5. All the matches which have matches greater than equal to 4 are considered to be good matches.
6. Now comes the most important part of the image stitching, i.e. calculating Homographic matrix. After this matrix is calculated, both of the images are projected onto the same surface and are then blended together.

After these steps are completed, the resulting image is stored and displayed.