Part 1

The sample image was read and the RGB color values for each pixel were put into arrays to implement the k-nearest-neighbors algorithm. The original image is shown below.

Figure 1: The original image

Euclidean distance is utilized to assign centroids to their closest samples from the image. The measure for Euclidean distance is given below:

$$D(x_1, x_2) = ||x_1 - x_2||^2$$

The centroids were initialized randomly, and the iteration was stopped when the change in the clustering error is below 0.00001. To speed up the KNN algorithm, normalization was done on the sample image.

The algorithm was run with different values of k = {2, 3, 4, 5, 6}. The time taken by each execution is given below in Table 1.

K value	2	3	4	5	6
Time taken	0.634s	1.243s	1.780s	1.921s	2.026s

It is possible to see that the complexity increases as k value gets larger. The clustering errors for each k value is given below in Table 2.

K value	2	3	4	5	6
Error	0.611	0.488	0.391	0.338	0.303

As the number of centroids in the KNN algorithm increases, the clustering error decreases. There is a tradeoff between computational time and clustering error.

The mean vectors of clusters were used to represent each pixel in the image. For different values of k = {2, 3, 4, 5, 6}, the clustered images are shown below.

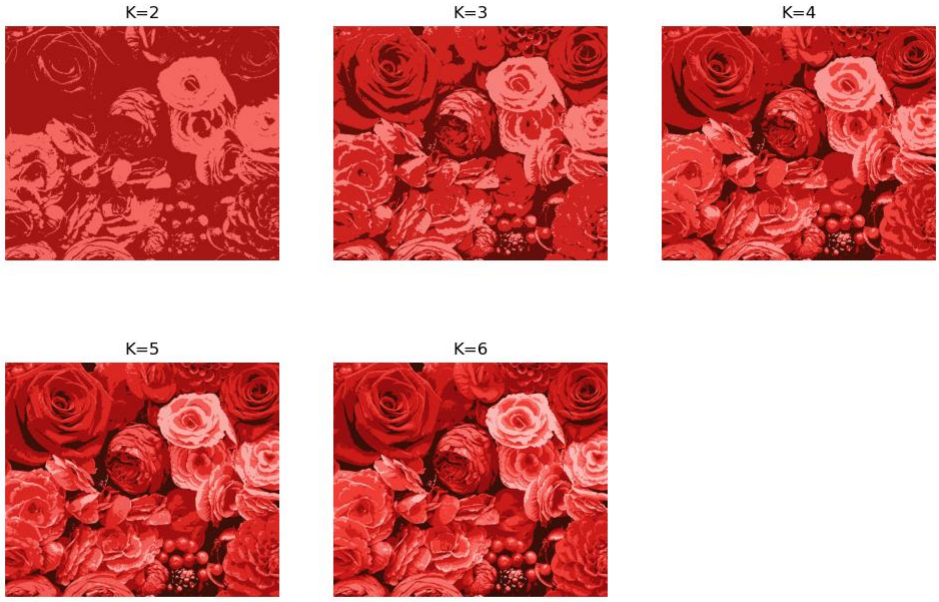

Figure 2: The clustered images with different k values

As can be seen from Figure 2 that the clustered images become closer to the original image as the number of centroids representing their pixels increases. In the clustered image with K=2, there are only two centroids to represent any pixel, that is to say, only two pixels are used to recreate the image. However, in the clustered image where K=6, the number of centroids was larger than 2, so the resolution of the recreated image is higher.

Later, different values of k = {2, 3, 4, 5, 6, 8, 10, 16, 32, 64} were tested for selecting the best k. The clustering error for each implementation was recorded and plotted on a curve after normalization.

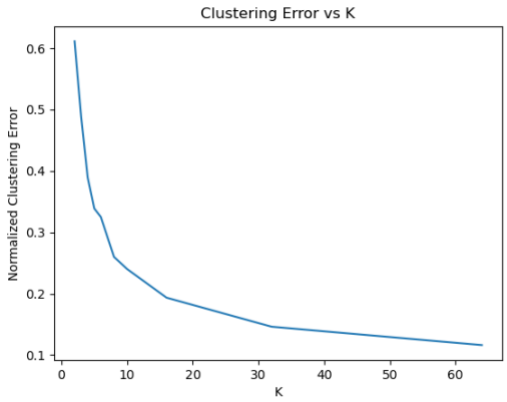

Figure 3: The clustering error over different k values

In Figure 3, it is observed that the clustering error declines as k increases, and that the value k = 15 is where both the error and the complexity are low enough. It is also seen that around k = 15 there is an elbow point, meaning that the enough variance is detained at that point.

The clustering error and running time when k = 15 are 0.198 and 8.512s, respectively. The clustered image is shown below.

Figure 4: The clustered image with k = 15

It is possible to observe that the clustered image where k = 15 and the original image look alike. This way, the image takes less space but represents well the original picture.

Part 2

In this part, single linkage and complete linkage were used as distance metrics. The single linkage measure is the minimum distance, and complete linkage measure is the maximum distance between clusters. To select one of the mentioned distances, the clustered images were shown with k = 15.

Figure 5: The clustered images with single and complete linkage

It can be seen that the complete linkage works with a much more performance, so it was chosen.

The centroids were initialized randomly, and the iteration was stopped when the change in the clustering error is below 0.00001. To speed up the AHC algorithm, normalization was done on the sample image.

The algorithm was run with different values of k = {2, 3, 4, 5, 6}. The time taken by each execution is given below in Table 1.

K value	2	3	4	5	6
Time taken	515.6s	589.5s	593.5	495.7s	761.8s

The clustering errors for each k value is given below in Table 2.

K value	2	3	4	5	6
Error	0.707	0.545	0.477	0.408	0.433

As the number of centroids in the AHC algorithm increases, the clustering error decreases.

The mean vectors of clusters were used to represent each pixel in the image. For different values of k = {2, 3, 4, 5, 6}, the clustered images are shown below.

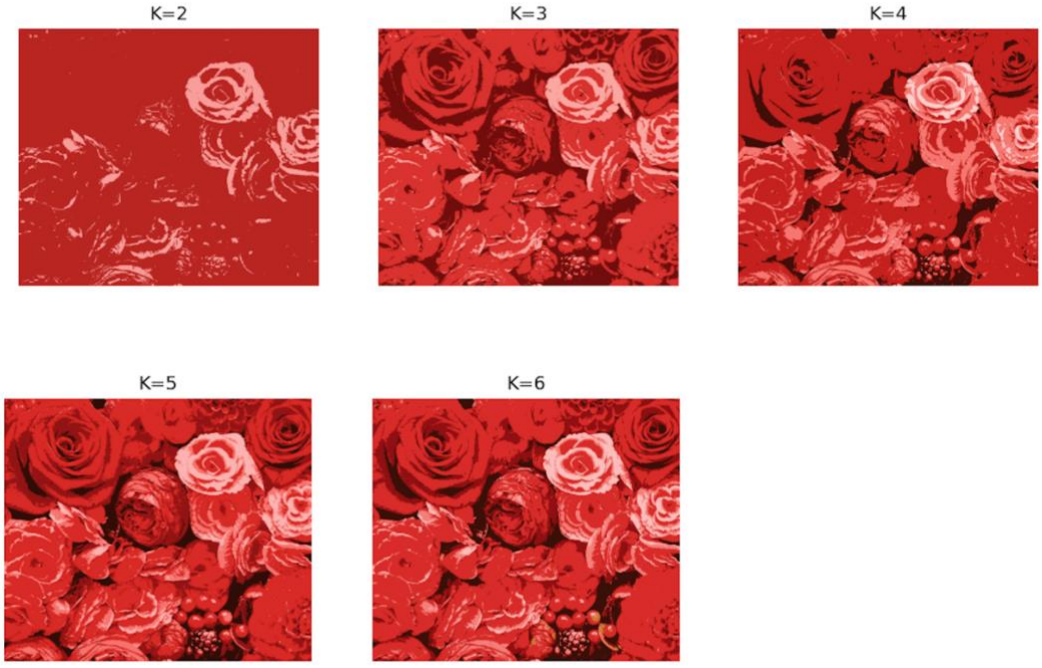

Figure 6: The clustered images with different k values

Later, different values of k = {2, 3, 4, 5, 6, 8, 10, 16, 32, 64} were tested for selecting the best k. The clustering error for each implementation was recorded and plotted on a curve after normalization.

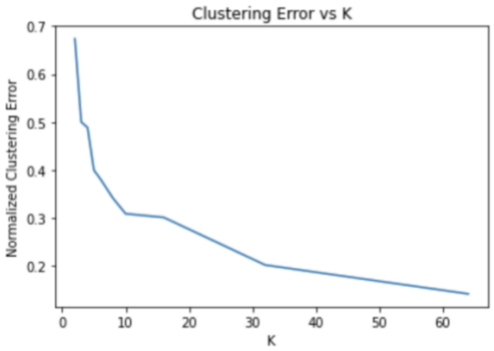

Figure 7: The clustering error over different k values

The value k = 32 is where both the error and the complexity are low enough. It is also seen that around k = 32 there is an elbow point, meaning that the enough variance is detained at that point.
The clustering error and running time when k = 32 are 0.202 and 193.4s, respectively. The clustered image is shown below.

Figure 8: The clustered image with k = 32

It is possible to observe that the clustered image where k = 32 and the original image look alike. This way, the image takes less space but represents well the original picture.

To overcome the computational complexity of the Agglomerative Hierarchical Clustering algorithm, KNN was used to initialize clusters from groups of pixels, then AHC was performed.

To conclude, KNN algorithm overperformed the AHC algorithm, because of the computational time and the clustering error. AHC needs more centroids and more time to yield a similar clustering error.

Gün Kaynar
22101351