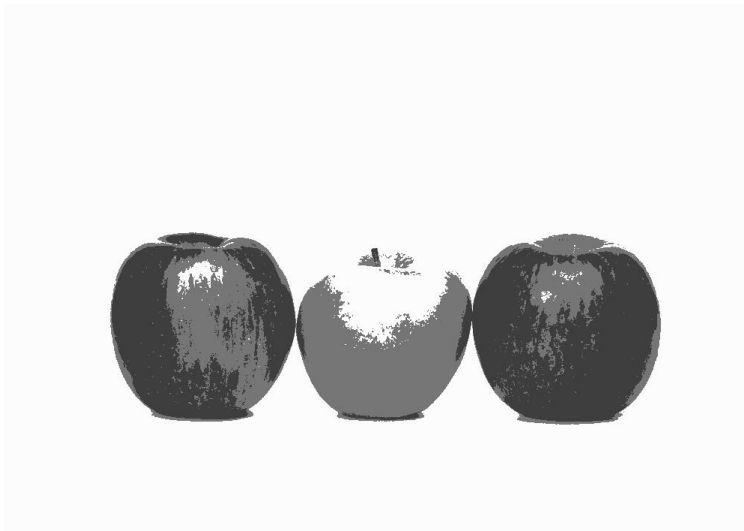GREYSCALE:
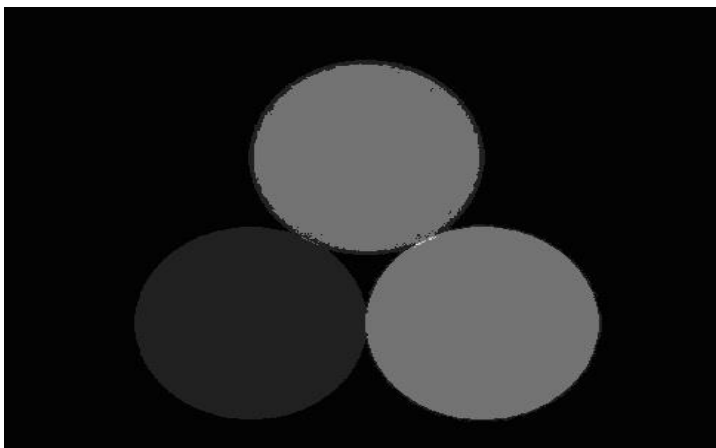
In the greyscale segmentation,

a. I took random centroids or intensity based on number of clusters that need to be segmented.
b. Used random function to generate coordinates in the image and took the intensities at that coordinates.
c. Traversed each pixel in the image and calculated the intensity differences with all the centroids.
d. Based on the intensity difference, assigned each pixel to the cluster in a temporary array with lesser difference.
e. Calculated the mean of each cluster with the intensity and number of pixels belong to that cluster.
f. Removing clusters, if no pixel belongs to that cluster.
g. Run steps from c to f till old mean is equal to new mean, that till the cluster centres converges.
h. Assign each cluster's intensity to image pixel intensity by looking in to the temporary array.
i. Return the image

K= 3 (fruits.jpg)

K= 4 (circles.jpg)

K=2 (Kenny.JPG)

RGB:

In RGB segmentation,

a. I took random centroids or intensities of RGB based on number of cluster that need to be segmented.
b. Used random function to generate coordinates in the image and took the RGB intensities at that pixel.
c. Traversed each pixel in the image and calculated the Euclidean distance with centroid of the cluster.
d. Based on the distance, assign each pixel to the cluster in a temporary array.
e. Calculated the R, G, B means of each cluster with intensities and number of pixels belong to that cluster.
f. Run steps from c to e till the mean converges or number of iterations are less than 10.
g. Assign each cluster's intensity to image pixel by looking into the temporary array.
h. Return the image.

K=3 (fruits.jpg)
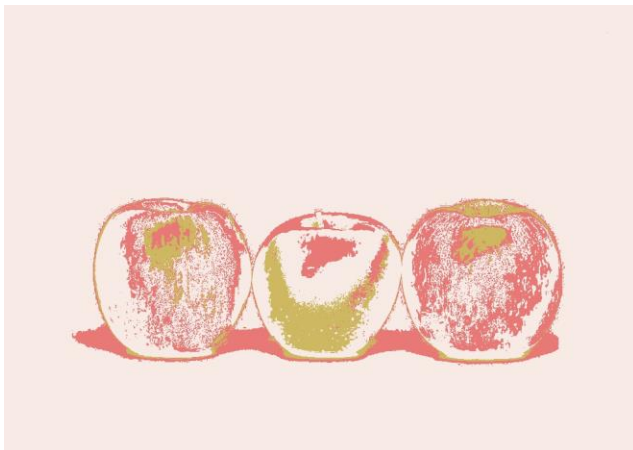
K=4 (circles.jpg)
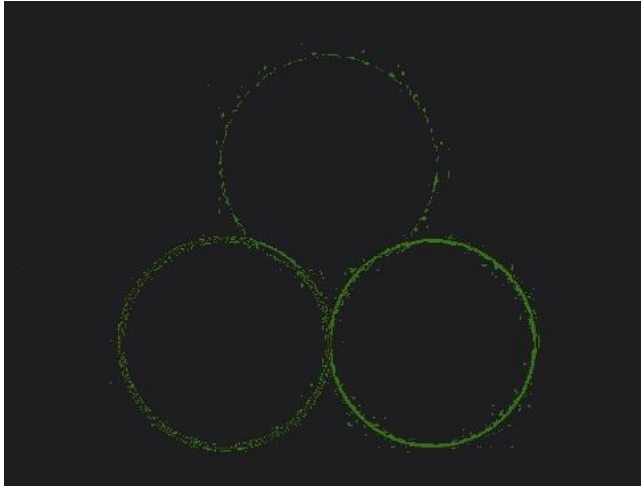
K= 2 (Kenny.JPG)

1. What are the parameters that influence your algorithm? Explain their effect?

K value effects the most of algorithm most and by taking random values for K has more impact on the output. As the K value increase the image looks more like the original image for both grey scale and RGB but as K increase the run time of the program also increase.

As I am taking **random points**, if all the random points belong to approximately same intensity, the image after segmentation no as expected.

Number of loops it takes to converge centroids, as grey scale image has only one intensity at the pixel, it takes less time to converges but RGB image takes more time. I made the RGB segmentation to run the program till the centroids converges or number of loops is less than 10. If I terminate the program with more iterations the image is more segmented.

Number of pixels effect the algorithm and time to execute the program, as high-quality image more number of pixels which are to be segmented and an image like RGB will increase it execution time with 3 times more than grey image.

2. Does your segmentation code always generate the same segments for a given k. Explain?

No, it doesn't return the same output for same K value because I am using random function to generate coordinates in the image to get centroids on the image. As the clusters are randomly assigned, it gives different output for each run even though for the same k value.

3. What is the objective of this implementations? Which set of features perform better (color vs greyscale)? Explain?

The goal is to find meaning from an image whether it is to identify an object, understand interactions, etc. To find meaning from an image, image segmentation makes everything easier than finding meaning from pixels.

Greyscale image performs better than RGB image as there are only one set of intensities in the greyscale, the segmentation is more sensitive towards greyscale than a RGB image. While in RGB images there are 3 set of intensities for each pixel, it is quite difficult to segment.

For larger K values, the image is more segmented, but the execution time also increases with it. In RGB images both the K value and number iterations till the centroids converges effect the performance of the program.