Alex Iacob

Prof. Kinsman

CSCI 431

October 19, 2022

***Conclusion:***

Overall, I used a very large majority of the code that Prof. Kinsman provided, but I think that I have an understanding of it. The entire functionality for clustering followed the general formula of “read in the image and separate the channels, prepare data to go into the kmeans method, format the results a little, then display them nicely.” The kmeans method used here uses four distinct parameters. The first is the red, green, and blue channels; this is the actual data that the algorithm is going to use to cluster “things'' with. The number of seeds used allows the kmeans algorithm to know how many clumps to cluster everything into. The other two arguments demonstrate each color’s starting point to begin the clustering process; in particular these were the 12 color namespaces. After the algorithm takes its sweet time and runs, we are left with displaying its results. After this the code does some weirder things, I have not seen that you can make a custom colormap before.

The next two questions also had their code given to us by Prof. Kinsman. Upon running it the first time, it seemed that the given code ran everything as expected. Like before, we started off by reading in the image and compressing it. We asked the user to click on all of the raspberries/oranges first, then click on a bunch of things that were **not** raspberries/oranges. We then tried to determine whether any one particular pixel belongs to a cluster or not by using the mahalanobis distance. If the given pixel is closer to the foreground color, then it is classified as being in the foreground (a raspberry/orange), otherwise it belongs to the background. This process is repeated for every pixel in the image and we are left with a masked image. We then turn this result into a uint8 image and write it to our file system. It is also worth noting that depending on the amount of fruit you click, the cleaner the image looks. This definitely makes sense because we are removing ambiguity from the clustered points.

**Raspberries:**



**Oranges:**

