

## HW07

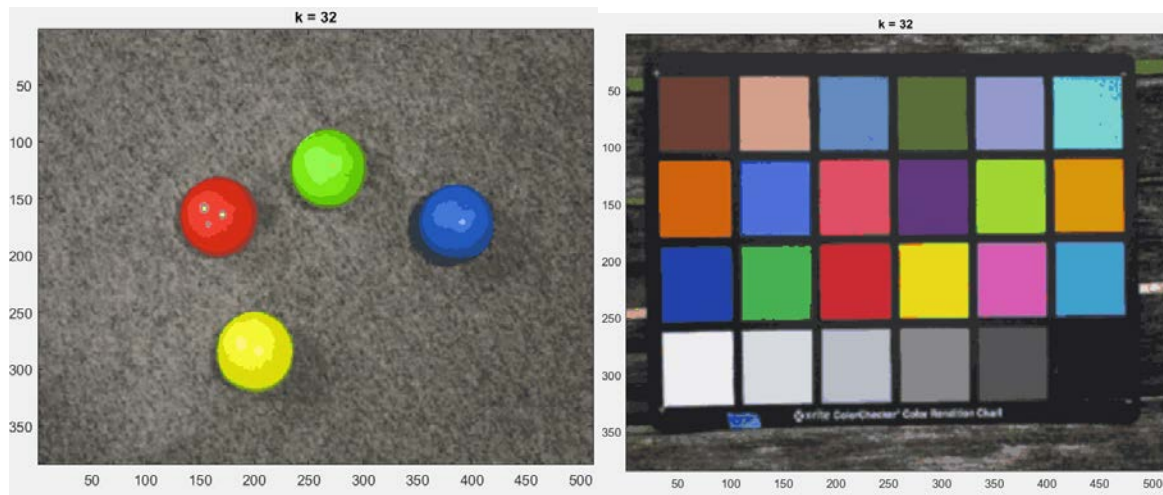
### 1. Overview

The goal of this assignment is to write a good cartoonization algorithm such that it preserves region details well enough to be able to see what the subject is, but at the same time does not provide too many details.

### 2. Parts

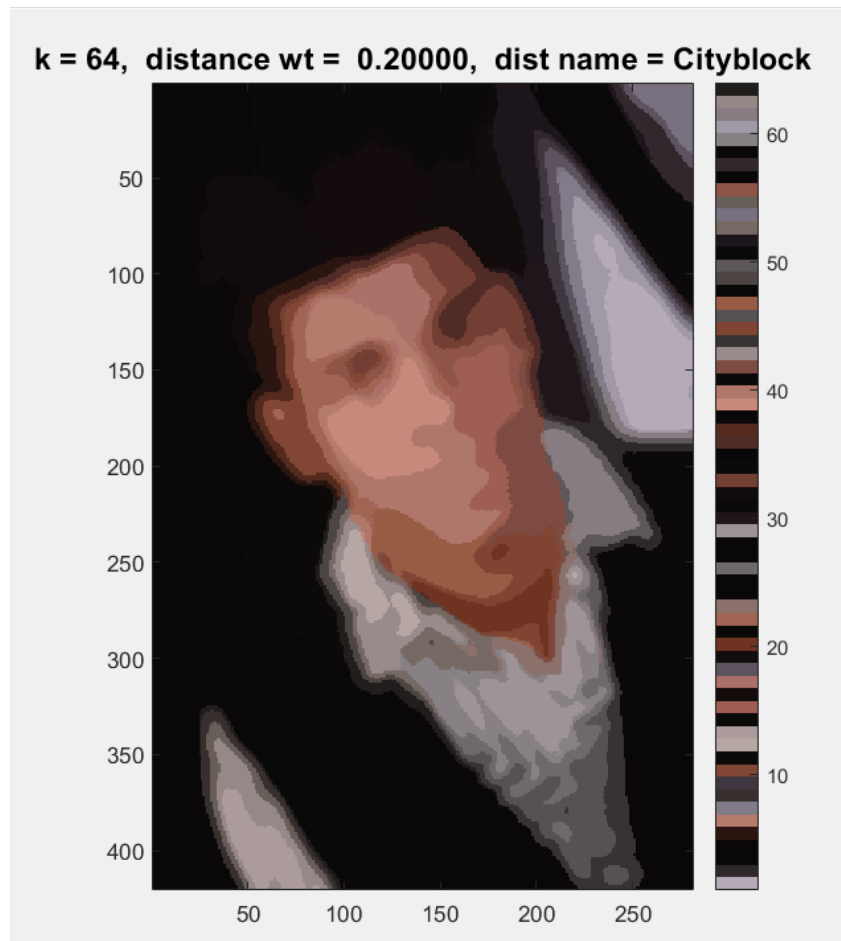
#### 1) HW07\_part2\_Changing\_K.m:

I started this part with some noise removal using a Gaussian filter of size 5 and sigma 1. The best value of  $k$  for both images was at  $k = 32$ .



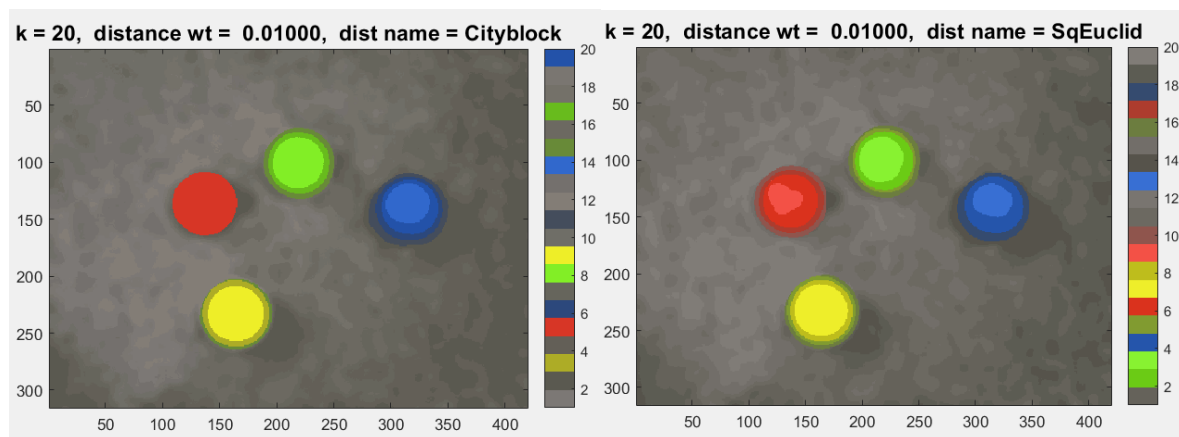
#### 2) HW07\_part3a\_DistanceWts.m:

For this part, I performed K-Means in the YCbCr color space. The distance weight that worked best for me was 0.2. For values below 0.2, the image didn't have the desired cartoonization. For values above 0.2, the image was more cartoonized and took a longer time.



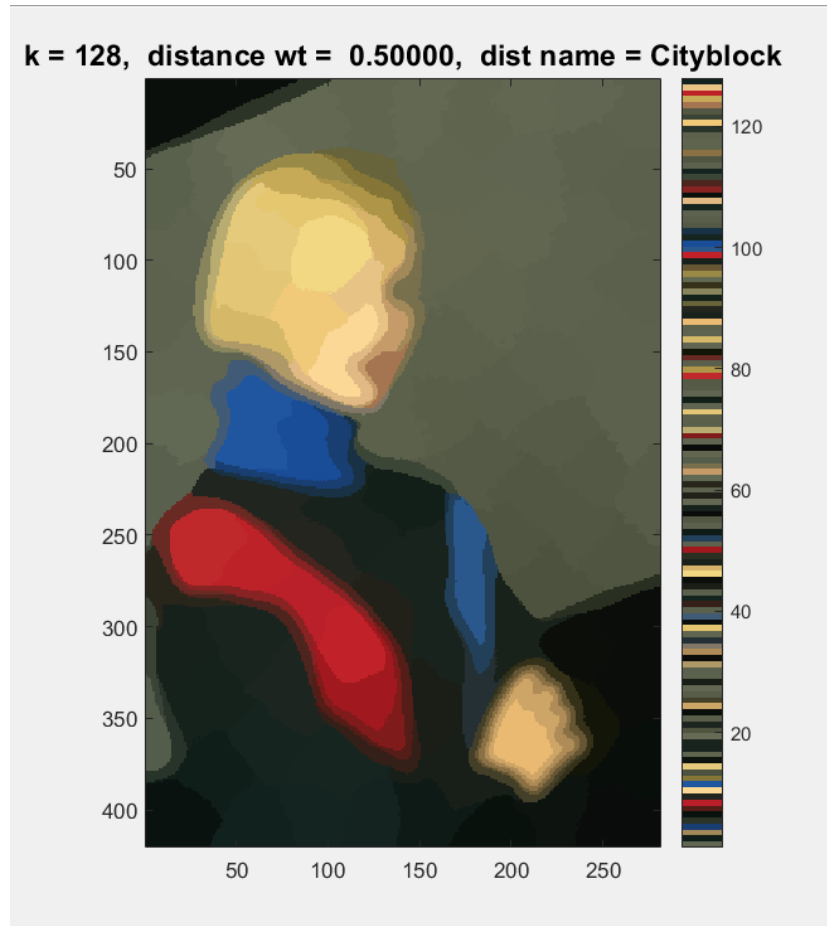
### 3) HW07\_part3b\_ Euclidean\_vs\_CityBlock.m:

For this part, I performed K-Means in the YCbCr color space. The distance metric 'SqEuclid' gave better cartoonization and took less time to execute than 'Cityblock'.



#### 4) HW07\_part3c\_198023.m:

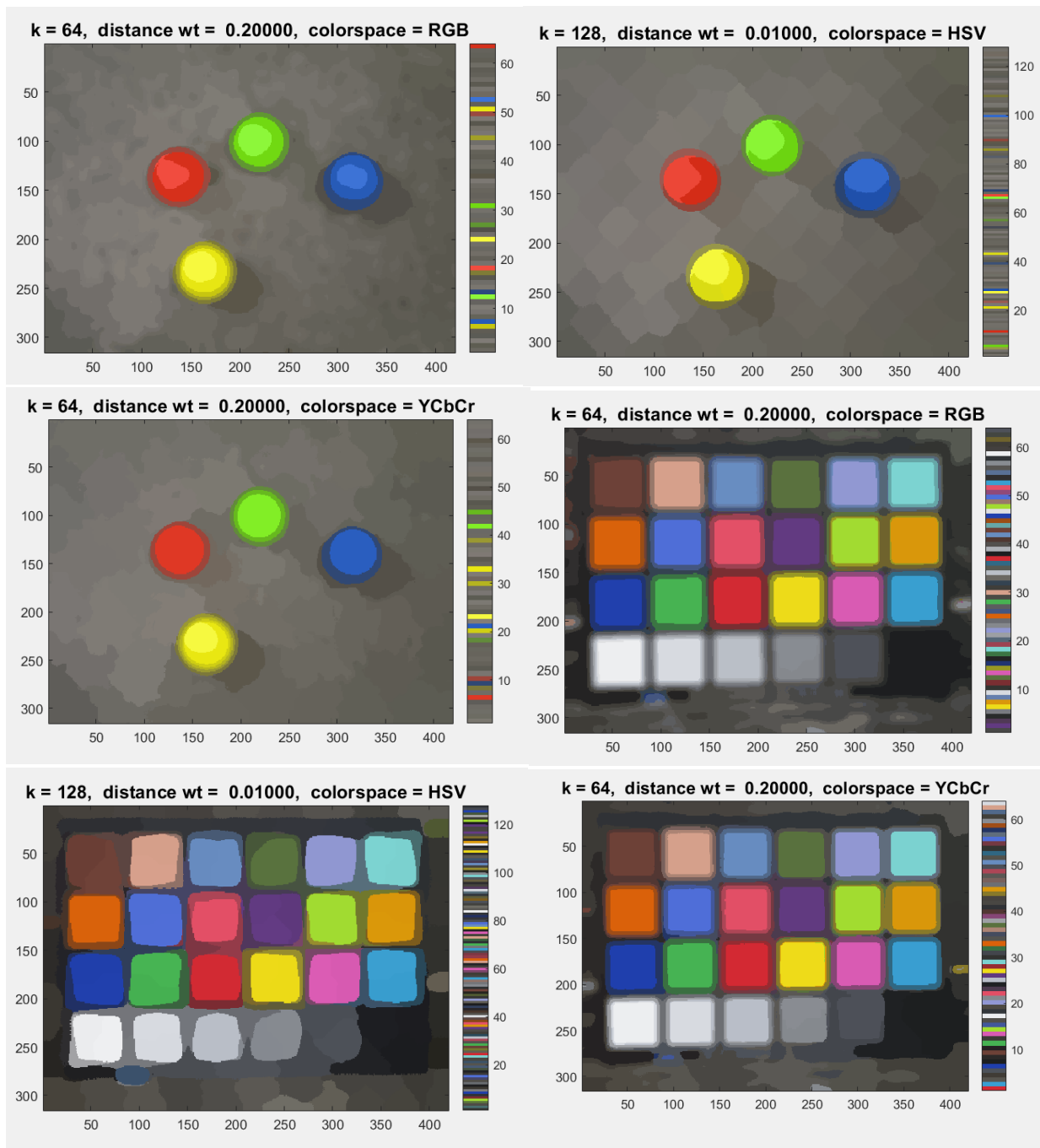
For this part, I performed K-Means in the YCbCr color space with a distance weight of 0.5 since it seemed to work best for this color space and no. of clusters = 128. I also used the distance metric 'Cityblock' since 'SqEuclid' took longer to execute.

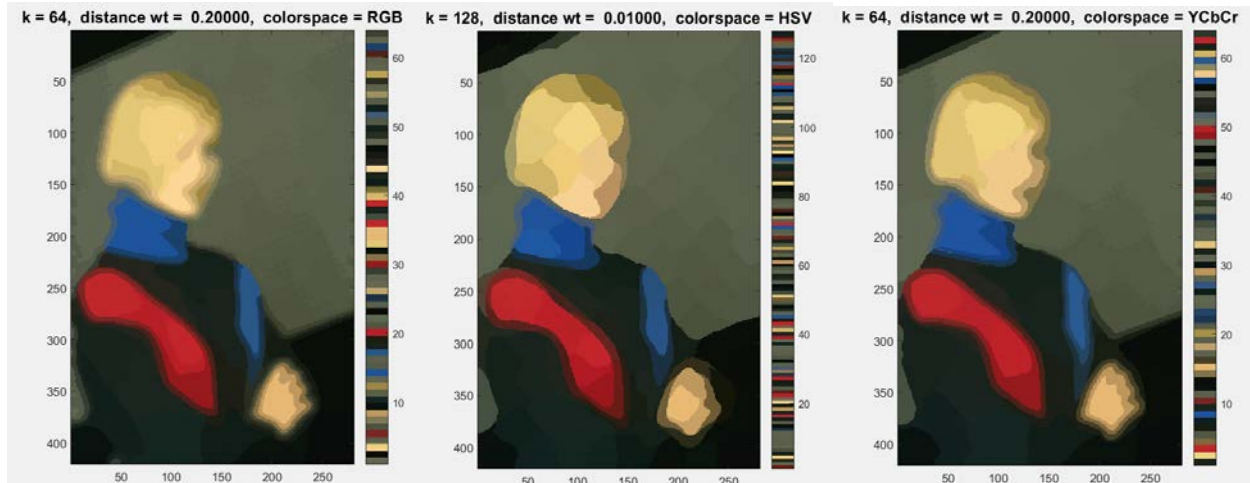


#### 5) HW07\_part3d\_Color\_Spaces\_Impact.m:

In this part, I observed that RGB was the fastest for the BALLS\*.jpg image while YCbCr was fastest for the other two images. Both these color spaces gave almost identical results for the same no. of clusters, distance weight, and distance metric. The HSV color space took the longest amount of time among all three color spaces and also required higher number of clusters to give a good cartoonization. Overall, the RGB color space gave the best results for cartoonization while still maintaining boundaries between objects.

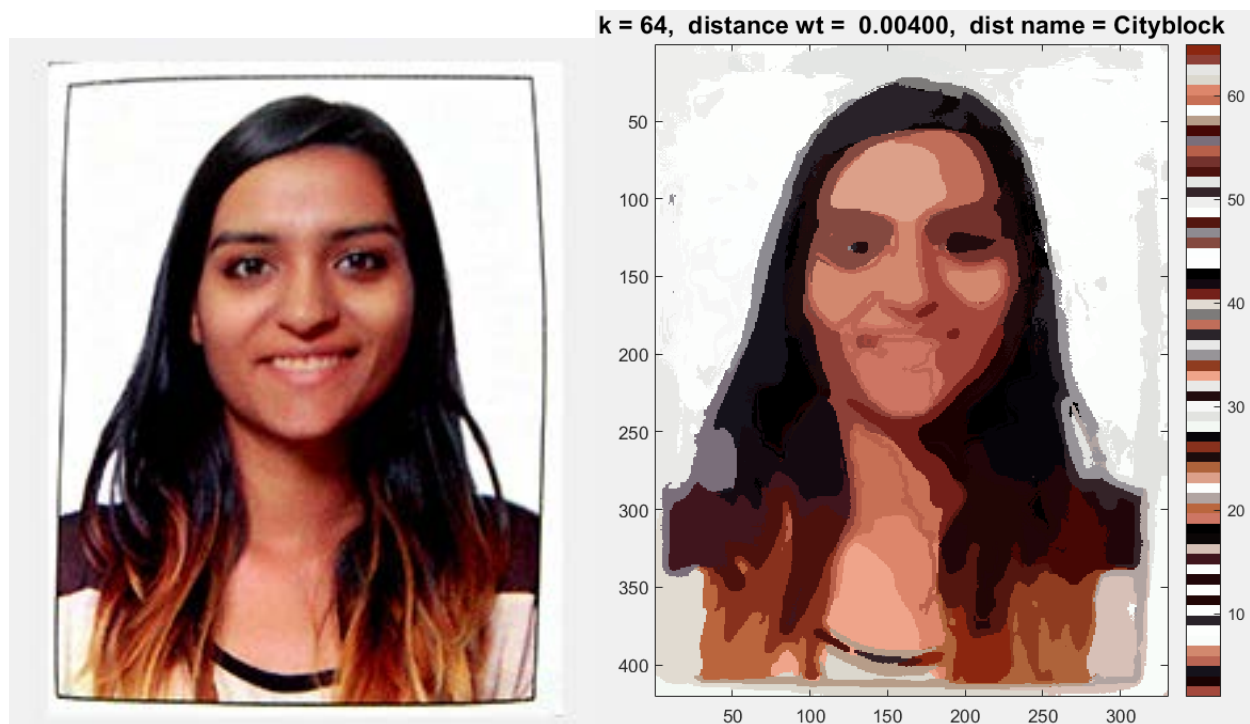
The Lab color space gave similar results to YCbCr but took a lot longer to execute.





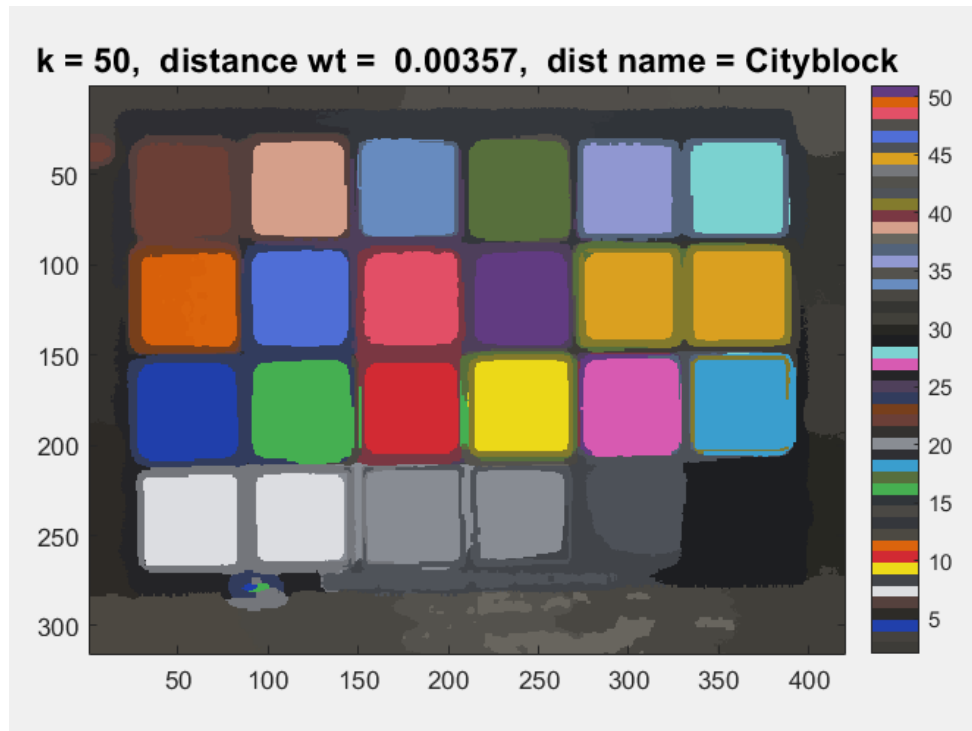
#### 6) HW07\_part4\_portrait.m:

In this part, I first detected the edges in the portrait using the Sobel edge detecting operators. I then blurred the image using a Gaussian filter with a sigma of 5 and the edges using a Gaussian filter with a sigma of 1. Then, adding the edge strength to the attribute list, I perform K-Means using  $k = 63$ , a distance weight of 0.004 and the distance metric 'Cityblock'. Finally, I added the blurred edges back to the cartoonized image.



### 7) HW07\_part5\_Edge\_Strength\_Bonus.m:

In this part, I observed that adding edge strength to the list of attributes helped decrease the number of clusters as well as the distance weight.



### 3. Conclusions

In this assignment, how the `rgb2ind()` function in Matlab can be used to create a color palette and in cartoonization.

I've also learned to write a cartoonization algorithm and how to use the `kmeans` function in Matlab. I've learned how different values of  $k$  (number of clusters) affects the final result of the cartoonization. The value of  $k$  depends highly on the pictures but also on the color space being used, distance metric and other parameters.