Mathew Seedhom
CS 558
Homework 2
I pledge my honor that I have abided by the Stevens Honor System.

1. kMeans

   For part one of the homework we had to implement the kMeans algorithm with k = 10. In order to do this I first had to initialize 10 centers, and then recursively cluster pixels solely based on their colors. In order to do this I compared the distance of every pixel to every center, and put the pixel in the cluster of the center that was the closest. In order to optimize the time complexity, instead of using a square root to calculate the Euclidean distance between the colors, I added the squares of each of the differences of colors, and compared the minimum of the distance squared. Then I had to compute the average color of the rest of the cluster. To optimize this, I simply added the value of each color component and stored it, and then divided that number by the number of pixels to contributed to those sums, rather than computing the average after clustering each point. I also floored these averages to optimize time complexity (it can be thought of as a threshold of 1). From this, the new center colors are calculated and I check to make sure if there was no change. I also made the function print time stamps for different segments in the code to ensure there was no unintended time performance issues with the code. I initially tested the function applying the function to the image with 2 centers.

2. SLIC

   For part two of the homework we had to implement the SLIC algorithm with segmentation distance of 50. I fist set the centers to every coordinate non-zero multiples of 25 as the x and y coordinates to be the cluster centers. I then readjusted the centers to be the pixel with the lowest total gradient from the adjacent or corner pixels to the original centers. At this point I assigned pixels to clusters in a 100*100 area around each center with a minimum Euclidean distance between each color component and their pixel coordinates, while weighing the pixel distance down by 4 (1/2 squared as requested). I applied the same method I did in part 1, and did not square root the distance. At this point, I averaged the colors and the pixel distances in the same way as earlier to compute the new centers, and repeated the process until convergence in the same way. When doing this I noticed the clusters did not look similar to those in the class presentation, so I further weighed the pixel distances by a factor of 100 for a better result.