CSE 6740
Assignment 1
Zheng Cai

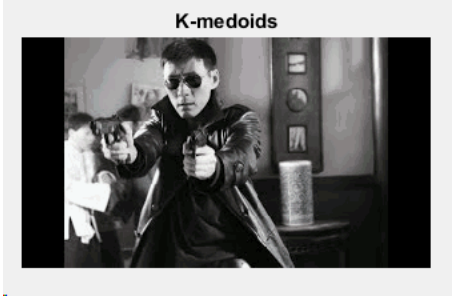
1.

In the K-medoids framework, I initialize centroids by randomly selecting K different points from 0 to 255, and then changing centroid by another point which is in the cluster of this centroid to see if dissimilarity reduces. If it works, replace it with centroid and execute clustering again. It will stop until all centroids become stable or specific rounds of loop finish. Rather than Euclidean distance in K-means, Manhattan distance is used in my K-medoids.

2. Original picture is :

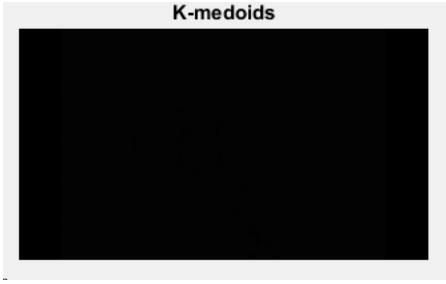
Here is the picture when I implement k-medoid with k=3:

Then I increase the k to 50, and that picture turns to:



For the K-medoid, we can see that lager K leads to a more clear picture which is like the original one because it is not compressed much. Although pictures' quality is quite similar it K is small, K-means will compress more clear and more colorful picture than K-medoid when K becomes large. The running time for k=3 is 0.269s and it goes to 2.008s when K=50. But for K-means, it is much longer, it executes 0.716s and 18.565s when K=3 and 50.

4.



K=3 K=50

I intentionally set initial centroids equal to [0,0,0], [255,255,255] and [128,128,128], and later set them to [1,1,1], [2,2,2], [3,3,3]. Here are the outputs above. It impacts the results if there are compared with randomized implementation from color and clearness.