k-Means Clustering on Satellite Images

1. Initialize corresponding *k* color values for *k* clusters.
   e.g. *color* = [[0,255,255], [255,255,0]] for 2 clusters

2. Input satellite images, *imageInput* (512, 512,4).

3. Initialize *clusteredImage* as the output image.

4. Select k random x,y co-ordinates and extract pixel values from *imageInput* to initialize *initCentroids* (k,4), where each centroid has dimension (1X1X4).

5. *classData* = clustering(*k*, *centroids*)
   #*classData* will contain pixel information and assigned class information.
   Explanation of clustering(*k*, *centroids*)
   a. *classData* = classifyPixels(*k, centroids*)
      #use Euclidean distance to find distance between the pixel values of centroids and each pixel in *imageInput;* assign pixel to the centroid with minimum distance
   b. $centroids_{up}$ = updateCentroids(*classData*)
      #use *classData* to derive the mean of each clusters
   c. if stopping criterion is satisfied:
      i. return *classData*
   d. else:
      i. clustering(*k,* $centroids_{up}$)

6. For each pixel in classData,
   a. *clusteredImage*[*classData*->pixel] = *color*[*classData*->pixel->class]

7. Display *clusteredImage*

Stopping criteria:

1. If *centroids* and $centroids_{up}$ are same
2. Euclidean distance between the pixel values of *centroids* and $centroids_{up}$ is less than a threshold value
3. Number of updates crosses a threshold value

Evaluation of clustering - silhouette coefficient

1. Initialize silhouette array *s* (512,512).
2. For each pixel *p* in *classData*, s_p = calSilhouette(*p*)
   Explanation of calSilhouette(*p*)
   a. calculate the mean distance of the point from all the points in the same cluster.
      For example, if *p* is assigned cluster 1, then $a_p = d_1$ = mean(dist($imageInput_p$, $imageInput_q$)), where *q* represent the points assigned to cluster 1.
      Remember the distance is to be calculated based on pixel values, not positions.
   b. calculate the means of the distances from clusters other than the assigned cluster, i.e. $d_m = \{d_2, d_3, d_4\}$.
   c. b_p = min(d_m)
   d. $s_p = (b_p - a_p)/\max(b_p, a_p)$
3. silhouette_coeff = max(*s*)

Assignment 10

1. Implement k-means clustering on satellite images for k=2,3,4,5. Check what changes if we use k-medoids instead of k-means.
2. Plot silhouette coefficients for k=2,3,4,5

Outputs expected:

1. Clustered images with k =2,3,4,5.
2. Silhouette coefficient comparison plot for k =2,3,4,5.

Note: implement user defined functions for Euclidean distance.