Assignment 4

This exercise is part of the course assignment. **Deadline for the assignment 30.11.2022 at 23:59**

The topic of this assignment is image segmentation via k-means clustering. For this assignment you should return

- The files main4.m, k_means_cluster.m and get_ssd.m. Each file should contain your name and student number (of both students if you work in pairs).
- Your answers to the questions in the analysis part. At the end of the course, you should return a single pdf containing the answers to all questions of the assignments. The report should contain also your name and student number (of both students if you work in pairs).

NOTE! For this assignment you are **not** allowed to use the matlab built-in function kmeans or any other similar clustering functionality, you should implement the algorithm yourself!!

Coding part (5pt)

In this assignment you will implement the k-means clustering algorithm. The algorithm takes as inputs the number of clusters k and a matrix of data $X \in \mathbb{R}^{N \times D}$, where each row is a point in a D-dimensional vector space. The algorithm outputs an array of length N containing the labels for each point in X (e.g. if labels(3)=4, then X(4,:) belongs to the fourth cluster) and a matrix $C \in \mathbb{R}^{k \times D}$ containing the *centroids* of each cluster. Each point in X is assigned to the closest centroid using standard Euclidean distance. The k-means algorithm can be summarized as follows

1. Pick k random points from X as initial centroids.
2. Assign each point in X to the closest centroid, e.g. if the closest centroid to X(3,:) is C(2,:) then labels(3) = 2
3. For each cluster $i = 1, \ldots, k$ update the centroid to be the mean value, i.e.

$$C_i = \frac{1}{N_i} \sum_{j=1}^{N_i} X_j, \tag{1}$$

where N_i is the number of points in X belonging to the cluster i and X_j are the points in X corresponding to the cluster i.

4. repeat steps 2-3 until the array of labels does not change from the previous iteration (or until a maximum number of iterations is reached)

Your tasks for this assignment are

- Run the demo demo.p to get a taste of the assignment (might take a few minutes).
- Implement the above described k-means algorithm in the file k_means_cluster.m. Details about input and output are given in the file.
- When you are done, test your function running the first cell in the script main4.m
- Next we will use our k-means clustering function to perform image segmenation. This is mainly done for you, however you will have to fill a few blanks in the script main4.m. Particularly you will have to:
  - Reshape the image I of size $H \times W \times 3$ into an array X of shape $HW \times 3$ so that each row of X has the RGB values of a pixel. **Hint!** matlab function reshape
  - After clustering, reshape the array labels, into a matrix of size $H \times W$, make sure that the position of the labels in the matrix is coherent with the pixels in the original image
  - The next cell will plot the segmented image, assigning to each pixel the color of its centroid. This is already done for you.
  - Try different values of k and observe how this affects the result. For reference, the demo file uses 4 clusters.
- One metric to choose the optimal value of k is the *Sum of Squared Distances* (SSD) defined as follows

$$SSD = \sum_{i=1}^{N} \|X_i - C_i\|^2, \tag{2}$$

  where X_i is the ith point in X and C_i is the corresponding centroid. Clearly, the more clusters we use, the smaller SSD will be (think, what is the SSD when using as many clusters as we have points?), however the optimal value of k is the so called *elbow point*, i.e. the point at which the slope of SSD changes from steep to almost flat, i.e. the point when SSD starts to decrease slowly. Your task is to

write the function ssd = get_ssd(X, kmax), which takes a matrix of data points X as before and a maximum number of clusters k_{max}. For each $k = 1, \ldots, k_{max}$ your function should perform k-means clustering and compute the corresponding value of the SSD. The function should output an array of length k_{max} containing the SSD-values.

- When you are done, run the last cell of the main4 script, which will plot the SSD as a function of a number of clusters. Note, depending on how efficient your k-means implementation is, this might take a few minutes.

Analysis part (5pt)

Answer the following questions in your report

- Why does it make sense to choose the elbow point as optimal number of clusters? (2pt)
- Attach to the report the SSD-plot you obtained. What is the optimal number of clusters for the peppers image? Attach to the report also the image segmented with your optimal number of clusters. Is the optimal number of clusters reasonable in this sense? Why/why not? **Hint:** look at the colors in the original image, how does the segmented image change if you use one less cluster or one more cluster? (3pt)