Image Processing and Learning Methods

Assign 3–Clustering based image segmentation

Due: 11:30 p.m. Thursday, April 8th, 2021 (Cleveland Time)

**General Lab Instructions:**

1. Save solutions in appropriate m-files. Be sure to place semicolons wherever appropriate, to suppress unnecessary console output.

2. Please include comments (e.g., **your name and assignment number**) at the top of each m-file. **In your main file, place a message “-----Done for Question \*-----” followed by a pause command at the end of each solution, where \* is the question number (i.e., 1, 2, 3, etc.).**

3. **Please submit your zipped m-files via the Blackboard system. Please do not send any images.**

**4. 100 points as the full score + 20 bonus points.**

**1. [60 points]** **Clustering based image segmentation-1**

Read in the image (soccer.jpg) and save it in an array named **soccer.** Implement the **kmeans algorithm** to divide **inputIm** into **K** clusters. The prototype of this function should be:

**function[cluster\_map, center, dist] =** **MYKMEANS(inputIm, K)**

where **inputIm** is the original RGB color image, **K** is the cluster number, **cluster\_map** is a single-channel image with the same size of **inputIm** andeach pixel’s value in **cluster\_map** is its assigned cluster index, **center** is obtained cluster center which should be a **K**-by-3 matrix, **dist** is the average intra-cluster Euclidean distance divided by the average inter-cluster Euclidean distance in feature space.

Call the **MYKMEANS** function to cluster the image **soccer** into **K**=6 clusters and then visualize the returned **cluster\_map** with colors. For each pixel in the **inputIm** image, just use its R, G, B values as its 3-dimensional features for clustering. For example, if **K**=6, the **cluster\_map visualization** should be similar as the following visualization:



**Note that:** You are not allowed to use Matlab built-in function ‘**kmeans**’ or any other clustering functions or third-party clustering packages for this question, and violation will result in 0 point. After calling the **MYKMEANS** function, you get **cluster\_map**, then you need to visualize the **cluster\_map** using different color to show different cluster index, for example: yellow for cluster id=1, pink for cluster id=2, …, etc.

**2. [40 points] Clustering based image segmentation-2**

Re-do the work in Question 1, but call Matlab built-in function ‘**kmeans**’ instead of your ‘**MYKMEANS**’ function to get similar results.

**3. [20 bonus points] Automatically choose K in kmeans algorithm**

Call your ‘**MYKMEANS**’ function for multiple times from **K**=2, 3, …, 10:

**function[cluster\_map, center, dist] = MYKMEANS(inputIm, K)**

Then, compare the obtained **dist**, choose the minimum **dist** to determine the best **K.** Your code should output the following sentence in the Matlab command window:

**For soccer.jpg, dist is \* when K=2**

**For soccer.jpg, dist is \* when K=3**

**…**

**For soccer.jpg, dist is \* when K=10**

**-----------------------------------------------------------------------**

**For soccer.jpg, the optimal/minimum dist is obtained when K=\*.**