Medical Image Computing
Assignment 2

Introduction to Clustering Algorithms

Through this assignment, you will gain knowledge about various clustering algorithms, their specific applications, and familiarize yourself with built-in functions designed for executing these tasks. You will delve into the analysis and comprehension of the diverse parameters necessary for these functions and how modifying these parameters influences the output. Additionally, you will acquire insights into image segmentation techniques.

Guidelines

1. Submit all of your code and results in a single zip file with name **RollNumber_02.zip**
2. Submit single zip file containing
    a. **RollNumber_02.ipynb** (single python notebook having each function in a different cell)
    b. **Results.** (Folder containing resultant images)
    c. **Report.pdf** (Report should contain your findings and things you learned)
    d. **Dataset** (Folder containing the images that you will be used in this assignment. Fourteen images are given to you and you should choose **5 images of your choice** to apply each clustering algorithms.)
3. Email instructor or TA if there are any questions. You can discuss with each other's but are not allowed to look at each other's code.
4. You can use google colab for writing the code and then you can also download it as .pynb for submission. https://colab.research.google.com
5. Not following all Guidelines and Naming conventions will result in serious penalty.
6. **Deadline for submission of this assignment is Tuesday, 31th October.**
7. **A deduction of 10 marks will be applied for submissions made within 1 to 24 hours after the deadline. If your submission is delayed by more than 24 hours, a deduction of 50 marks reduction in marks for the assignment will be enforced.**

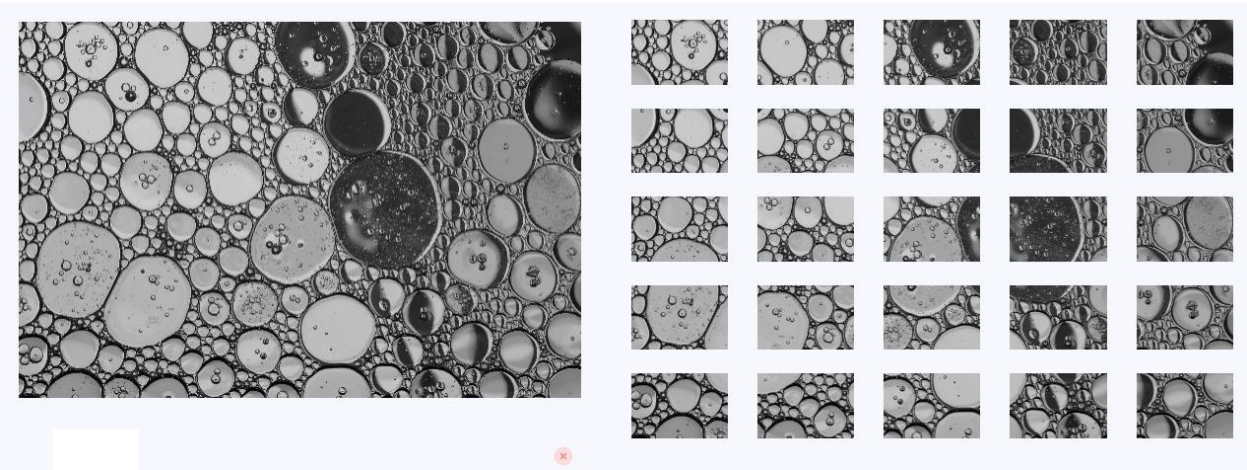
Data preprocessing:

Choose any 5 images of your preference from the provided dataset folder. Subdivide each image into separate patches based on the specified dimensions for each image as follows:

- **A.** 1st image = 5 x 5 patches
- **B.** 2nd image = 6 x 6 patches
- **C.** 3rd image = 10 x 10 patches
- **D.** 4th image = 12 x 11 patches
- **E.** 5th image = 20 x 15 patches

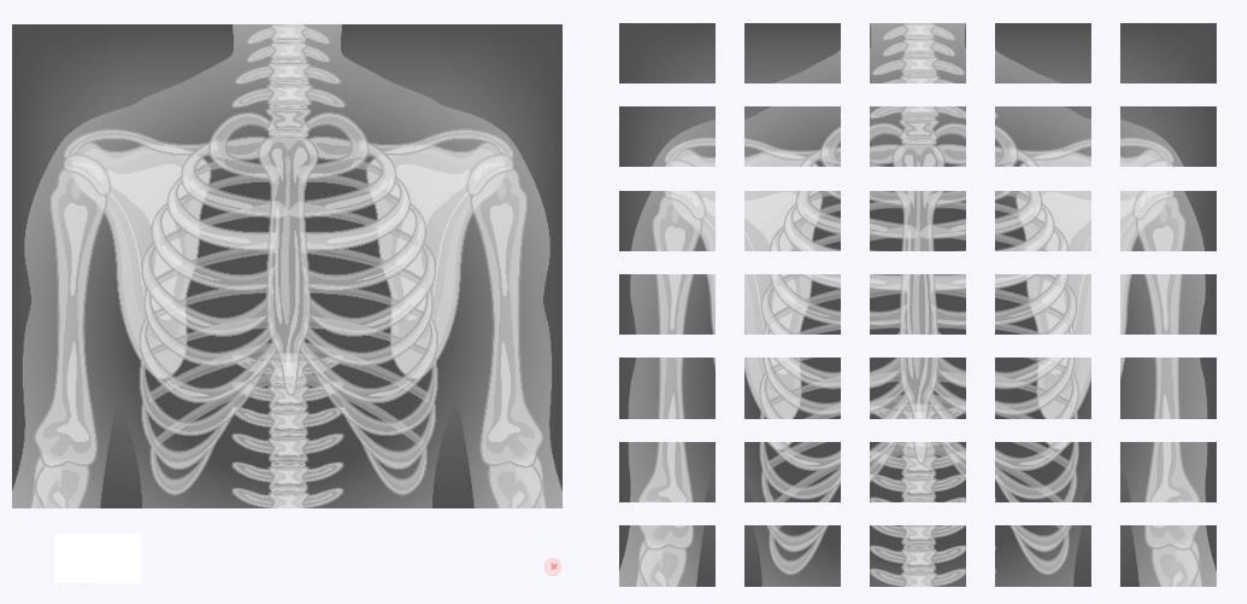
Compose your code to accomplish this task independently. You can use any build-in library of Python to do these task.

An examples of dividing an image into equal and unequal patches:



Original image

Equal Patched Image: 5 x 5 patches

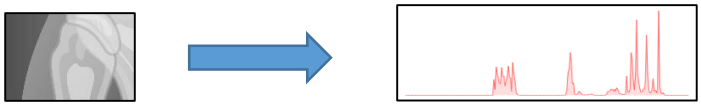


Original image

Unequal Patched Image: 5 x 7 patches

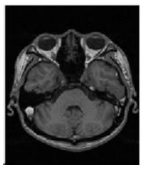
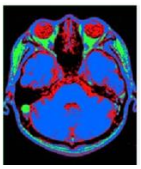
Tasks to be performed:

1. Make histogram of each image patch. Write your own code to do this task. You can use any build-in library of Python to do these task.
   Example:



2. Apply K-Means clustering algorithm on the aforementioned patched images, and utilize your roll number to ascertain the suitable value for k. If your roll number is even, opt for an even value for k; if it's odd, choose an odd value for k.

   Apply K-means clustering to each image patches. Every image patch will be assigned to one of the clusters. Assume, you choose K= 3, you assign patches that belongs to first cluster the red color, the one which belongs to second cluster blue color and one which belongs to 3rd cluster, green color. After that display these colors on the images.
   Finally, display the resulting images. Subsequently, analyze and discuss your results.
   Example of color-coded images are given below. Note that in your case colored image will not look so nice as you are performing patch-wise coloring instead of individual pixel-wise.

3. Apply mean-shift algorithm on above patched images. Apply mean-shift algorithm to image patches. The display the colored image as discussed above. Subsequently, analyze and discuss your results.
4. Select 5 images from the given 14 images and apply super pixel segmentation on it. Compute histogram for each of the superpixel. You will select the number of super pixels, either 50 or 100. Apply kmeans clustering and display colored image as discussed above in **1.** Display the resulting images. Provide an analysis of the results in your report.
5. Apply active contour segmentation on the five images and then display the resulting images. Subsequently, display your results and engage in a discussion about your findings.
6. Select 5 images from the given 14 images and apply Otsu's thresholding algorithm to transform each image into a binary format. Ensure that you provide a detailed description of the procedure in your report.

What to submit:

Submit the following in a single zip file as mentioned above.

1. Jupyter notebook file containing each task in a separate cell. Named as RollNumber_02.ipynb.
2. Resulting images against each task. ImageName_TaskNumber. (in Results folder)
3. Dataset
4. Report. Write detailed report on each task. Discuss what setting you chose and what outcome did you get and discuss how changing parameters affected your results.