CS 550, Spring 2023
Homework Assignment #3

Due: Wednesday, 06 June 2023, 23:59

Bonus assignment: this assignment will contribute to the total grade by 10% over and above the 100%. The grades of those students who do not wish to do this assignment will not be affected adversely. No late submissions will be accepted for this assignment.

Part 1

Implement the k-means clustering algorithm to cluster image pixels. Your algorithm should take the pixels of an RGB image, group the pixels into k clusters, and output the labels of the clustered pixels. In the RGB colour space, each pixel is represented with 3 values, which are between 0 and 255. To test your algorithm, use the image (sample.jpg) provided along with this assignment. First run your implementation for the following values of $k = \{2, 3, 4, 5, 6\}$.

For each of the above k values:

- Report the clustering error (averaged over all pixels)
- Report the computational time
- Provide the clustered image. To obtain this clustered image, represent each pixel with the RGB values of the corresponding cluster's mean vector.

Then, using your implementation, determine the optimum k value for this image. For that, do NOT just consider the k values given above (*i.e.*, do NOT just consider $k = \{2, 3, 4, 5, 6\}$). Explain how you determine the optimum value. Moreover, for your optimum k value, report the clustering error and give the clustered image.

In your report, give all details of your implementation, such as how to select the initial values and the stopping criterion. Your report should also include the aforementioned outputs. In this assignment, you may use any programming language you prefer. If you do not know how to process images in your preferred programming language, you can employ the following Matlab codes. Of course, if you do not want to use the following codes, that is completely ok as long as you process the image pixels and give the required outputs.

```
    % Reads sample.jpg into an im matrix, whose dimensions are N, M, and 3
im = imread('sample.jpg');
% Reshapes the 3D im matrix into a 2D matrix, called im2, whose dimensions are NxM
% and 3.  In this 2D matrix, each row corresponds to a pixel, and the columns correspond
% to red, green, and blue channels in the RGB colour space, respectively.
im2 = reshape(im,[size(im,1)*size(im,2) 3]);
   % You need to run the k-means algorithm on the im2 matrix.  Your k-means algorithm
% should output the clustering vectors (let's call them V) and the labels of the
% clustered pixels (let's call them cmap).  To implement the k-means algorithm,
% you can use Matlab.  Of course, you have to write your own code; you cannot use
% the kmeans built-in function of Matlab.  However, if you find Matlab too slow
% or if you prefer using another programming language (but if you do not know how
% to read images in your preferred programming language), you can write the contents
% of im2 into a file and read this file in your program.  Similarly, you can calculate
% clustering vectors V and labels cmap in your program, write them into files, and
% read them from Matlab to obtain the clustered image.  Suppose that your program
```

```
% outputs matrix V, whose dimensions are k and 3, and vector cmap, whose dimension
% is NxM. Also, suppose that your labels are between 1 and k.  Then you may use
% the following Matlab codes to produce the clustered image
cmap2 = reshape(cmap, [size(im,1) size(im,2)]);
M = V / 255;
clusteredImage = label2rgb(cmap2, M);
% Shows the clustered image in Matlab and writes it into a bitmap file
figure, imshow(clusteredImage) imwrite(clusteredImage,'output.bmp')
```

Part 2

Implement an agglomerative hierarchical clustering algorithm to cluster the image pixels. Similarly, your algorithm should take the pixels of an RGB image, group the pixels into k clusters using this agglomerative algorithm, and output the labels of the clustered pixels. The computational time of this part could be high due to the number of pixels in the image. Thus, propose a technique to overcome this problem. For example, you may "somehow" form small groups of pixels and consider each of these groups as the initial clusters of your agglomerative algorithm. However, downsampling an image to a lower resolution and running your algorithm on this downsampled image will NOT be accepted as a solution. Do not forget that there is not only one correct solution for this part. Thus, try to consider different alternatives. Test your algorithm on the same image (sample.jpg) and obtain the results for different values of k. Prepare a report for this second part similar to Part 1, also addressing the same questions given in the first part. Additionally, for this second part, give the details of the technique that you will propose to overcome the computational time problem.

Submit:

You are expected to write your report neatly and properly. The format, structure, and writing style of your report, as well as the quality of possible tables and figures, will be a part of your grade. Use reasonable font sizes, spacing, margin sizes, *etc.* You may submit either a one-column or a double-column document. The IEEE manuscript templates are preferred. In your report, do not give any screenshots. Your report should not exceed 5 pages.

Submit the pdf of your report along with the source code of your implementation as a single .zip file to Moodle before the deadline.