MAEG5720

Computer Vision in Practice

Assignment 3

Date: 29th, October 2021

Due Date: 12th, November 2021

Submission Guidelines

The folder you hand in must contain the following:

README.txt – contains anything about the assignment that you want to tell the TA, including a brief introduction of the usage of the code.

Code/ - directory containing all your code (**only .m files and image files allowed**) for this assignment, which is expected to have at least one .m file along with one .jpg or .png files.

Report.pdf – **only 1 document** showing the question number, result and comments.

Rename the folder as <your student ID>-Asgn2, and compress it into <your student ID>Asgn2.zip, and upload it to the blackboard system. (For example, 1155123456-Asgn2/ and 1155123456Asgn2.zip, pay attention to the name.)

Please read the guidelines CAREFULLY. If you fail to meet the deadline because of a submission problem on your side, marks will still be deducted.

The late submission policy is as follows:

- 1 day late: -20 marks

- 2 days late: -40 marks

- 3 days late: -100 marks

Pay attention to the format before. **10% deduction for every wrong format** (filename, function name, etc.).

MAEG5720

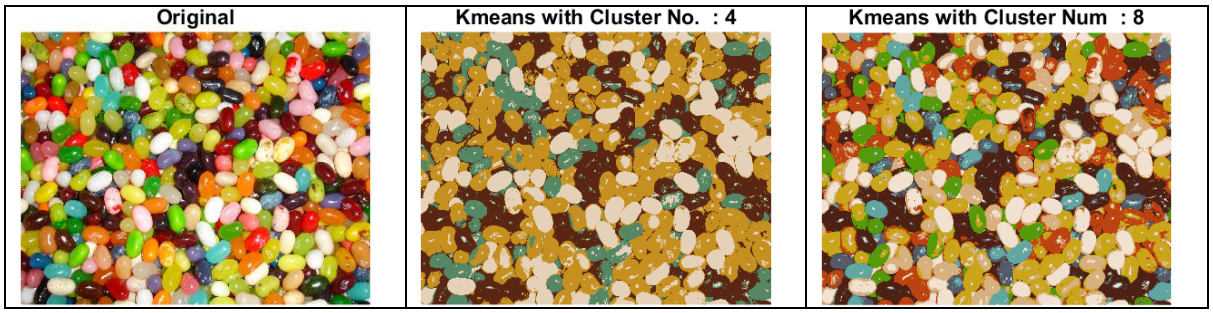
Computer Vision in Practice

Aim:

Images usually consist of many different objects. To better understand the image, we need to separate the objects for further processing. The process group pixels into different clusters with similar features, such as colours or spatial distance. This process is known as segmentation.

In this assignment, you will implement a well-known *k*-means algorithm. This is a general algorithm which groups the data into a set of *k* clusters, where *k* is defined by the user. One particular application of *k*-means is image segmentation in which the pixels with similar features (colours or spatial distance) are grouped into same cluster.

The following figures shows the results of *k*-means Segmentations



The k-means algorithm is easy to implement. The pseudo-code as follow:

Algorithm ***k*-means**

Input: RGB Images with $m \times n$ pixels. The user defined number of clusters, k
Max iteration number it_{max}

Output: A set of k labels, $\iota_1, \iota_2 \dots \iota_k$

1. Initialise k cluster centre $c_1, c_2 \dots c_k$
2. **do**
3. for each pixel p_i, **do**
4. $\iota_i = 0$
5. $dist_{min} = \infty$
6. for each cluster centre, c_j **do**
7. $dist_j$= distance of p_i to c_j
8. **if** $dist_j < dist_{min}$ **then**
9. $\iota_i = j$
10. $dist_{min} = dist_j$
11. **end if**
12. **end for**
13. **end for**
14. **for** each cluster centre, c_k **do**
15. $c_k = mean(p_i$ which label $\iota_i = k)$
16. **end for**
17. **end do** while $n < it_{max}$
18. change each pixel p_i to it's label c_k where $k = \iota_i$ for display

Question 1: Please write your own MATLAB code to implement the above algorithm

MAEG5720

Computer Vision in Practice

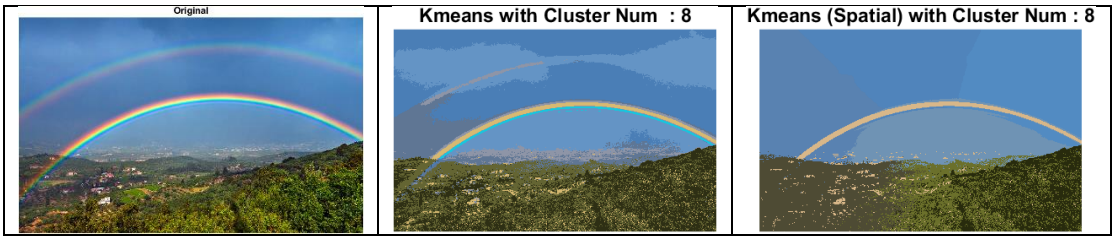
(a) Read an RGB image into your workspace
(b) Randomly Initialize the centre $c_1, c_2 \ldots c_k$
(c) Using the RGB value to compute the distance $dist(p_i, c_j)$ (30%)

$$dist(p_i, c_j) = \sqrt{\left(R_{p_i} - R_{c_j}\right)^2 + \left(G_{p_i} - G_{c_j}\right)^2 + \left(B_{p_i} - B_{c_j}\right)^2}$$

to perform the segmentation. Show the result for different number of clusters. (for instance, 4, 8). You can set the termination a fixed number of iterations.

(d) The termination criteria also another concern. You can do fixed number of iterations. Or you can do the following
   (i) After each iteration, check if ***each centre c_j changes*** within certain threshold value (20%)
   (ii) After each iteration, check if the ***number of points changes labels*** within certain threshold value. (20%)

Question 2: Besides RGB value, you can also put spatial distance into consideration.



The distance function can be easily modified as follow

$$dist(p_i, c_j) = \sqrt{\left(R_{p_i} - R_{c_j}\right)^2 + \left(G_{p_i} - G_{c_j}\right)^2 + \left(B_{p_i} - B_{c_j}\right)^2 + \left(x_{p_i} - x_{c_j}\right)^2 + \left(y_{p_i} - y_{c_j}\right)^2}$$

Please implemented the *k*-means segmentation with the above distance function. You can use termination method in either d(i) or d(ii). (30%)

What to submit: A pdf file with questions and answers. The .m files of the code. A readme.txt if you have additional information for TA.