**Rules: (Kindly read the rules very carefully.)**

1. Deadline for Project is: Practical Exam.
2. This project is intended for teams’ contribution. Cheating will **NOT** be tolerated. The penalty for cheating is a cancellation of the whole project mark, whether the incident happened on 1 or all the parts.

**Bonus**

1. Bonus 1: Make python GUI for your Project (Bonus+) (hint: you can use Tkinter or any other library)
2. Bonus 2: Detecting Automatically the suitable K value in K-means

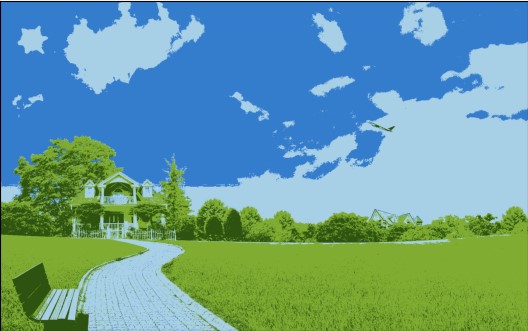
**Project Description**

**Part 1:**

**Apply Image segmentation on RGB images using K-means algorithm**

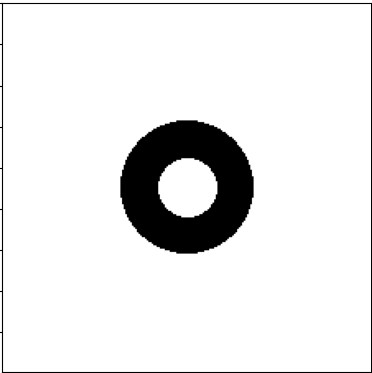
**Note: You can’t use a built in function that performs K-means , you must make all the code from scratch (including the calculations and equations)**

**The image must be RGB and you can’t convert it to gray scale , but you can convert it to any other mode such as HSV if you want.**



**Part 2:**

**Apply Band reject filter in the frequency domain to reduce periodic noise**

The band pass Filter looks like this in the frequency domain:

The image that you will work on is an gray scale image , but you should write a generic code that works with both gray-scale images and RGB , so it can works with other images

**Note: To obtain a good result as the example above , you should alter the radius of the inner and outer circles of the band pass filter to obtain a suitable result**

**Part 3:**

**Apply Histogram Equalization on a gray scale image , So if the image is RGB first convert it to gray scale and then apply histogram equalization on it.**

**Note: Don’t use built in functions to make histogram equalization such as cv2.equalizeHist() or ImageOps.equalize() or you will take zero in this part , you should implement the Histogram equalization function on your own.**

**A picture containing graphical user interface

Description automatically generated**

**Figure (a) is before applying histogram equalization and Figure (c) is after applying histogram equalization on the image , Figure b) and d) shows the distribution of histograms before and after applying histogram equalization**

**Part 4:**

**Make the following functions :**

* **A function that displays a histogram for a given image , if the image is RGB it displays three histograms one for each channel**
* **A function that takes two parameters a filter and its size , and apply the given filter on the image using convolution (this filter can be any filter it can be a smoothing filter or edge detection filter or any other filter) .**
* **A function that takes two parameters: the 1st parameter is a value , and the 2nd one is a mode either brightness or darkness , If the mode is brightness than the image pixels should be brighter with the given parameter value , and if the mode is darkness than the image pixels should be darker with the given parameter value**