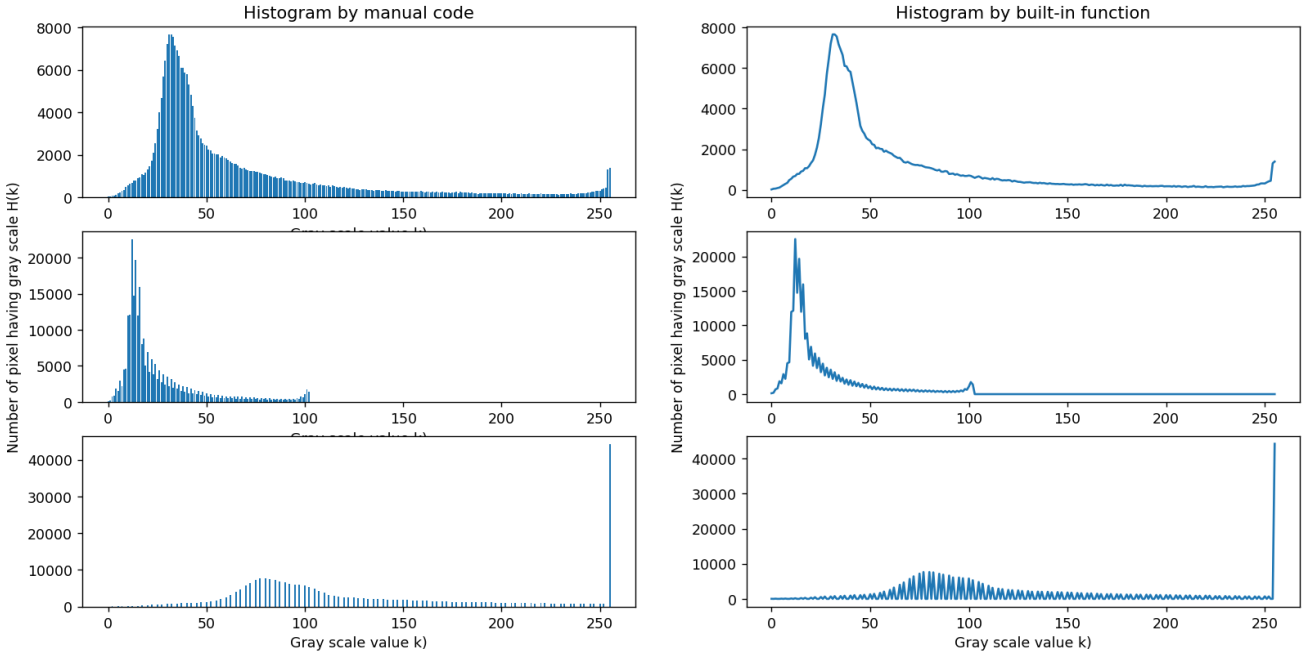
Name - Ajeet kumar yadav
Programme - M.tech
Stream- Signal Processing
SR no - 21117
Assignment -1

Note :

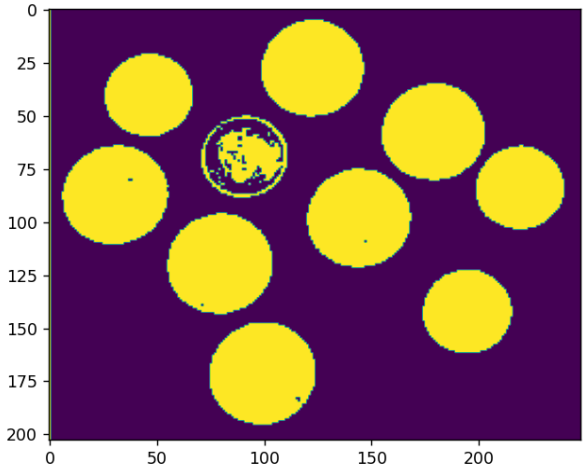
1. To run the code first extract the zip file and then open wrapper.py. After running wrapper.py it will ask you a one digit input from 1- 4
2. To run the first program enter 1 and for the second program enter 2 and so on.
3. I have used Visual studio code to write code, and the codes are working properly on VS code and may cause some errors on different IDEs having different versions of prerequisite.

Q.1 Output histogram graph through manual code(Left side) and histogram graph using opencv inbuilt function(Right side) for comparison. And curve variation is the same in case with manual code and by built in function of OpenCV.



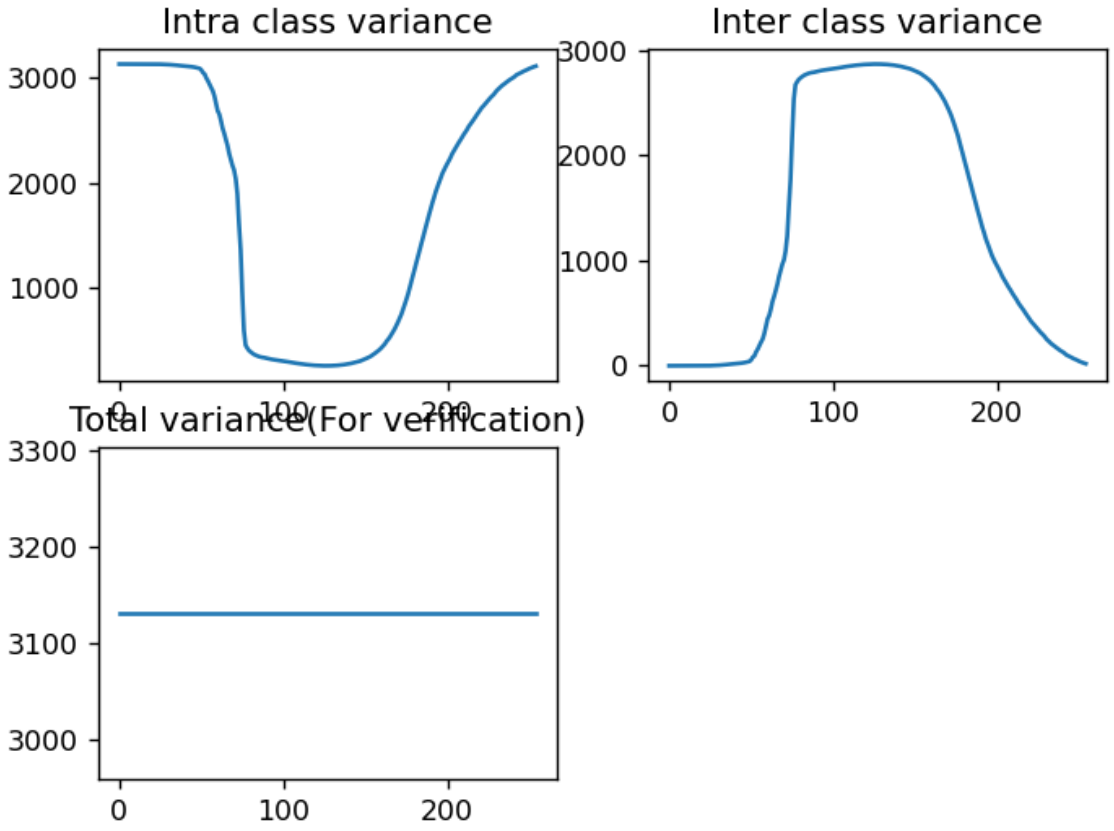
Q. 2 Values of intra class variance and inter class variance, and graph along with binarized image are as follows. And in graph we can see Lowest value in Intra Class variance and highest value in intra class variance are same lies on same

```
10  def otsus_binarization(image_gray):
11      within = []
12      between = []
13      total_variance = []
14      d =0
15      row, col = np.shape(image_gray)
16      # Histogram evaluation using first problem's function
17      x, histg = histogram_function(image_gray)
18      for i in range(255):
19          x, y = np.split(histg, [i])
20          x1 = np.sum(x)/(image_gray.shape[0]*image_gray.shape[1])
21          y1 = np.sum(y) /(image_gray.shape[0]* image_gray.shape[1]
22          x2 = np.sum([j*t for j,t in enumerate(x)])/np.sum(x)
```

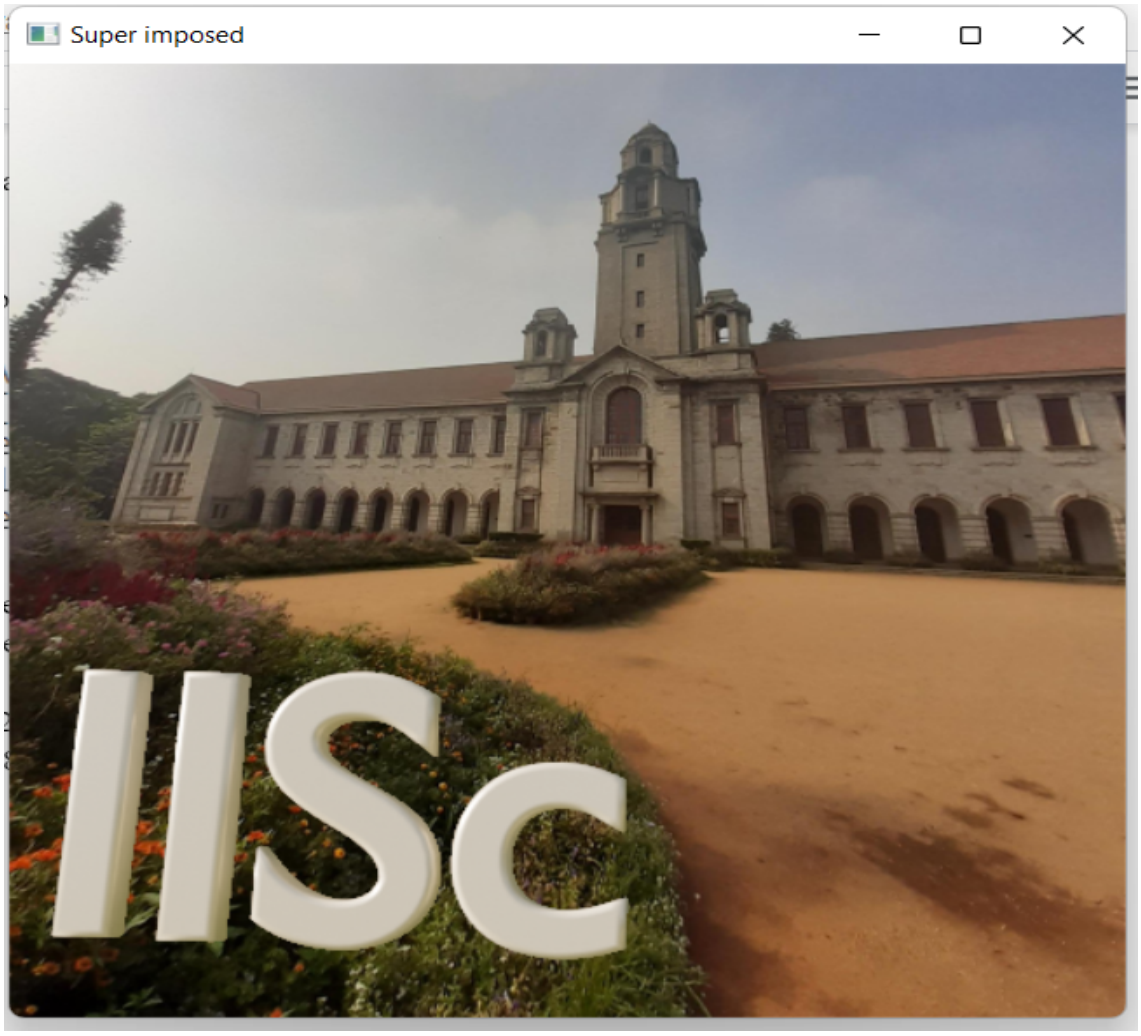


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
  x2 = np.sum([j*t for j,t in enumerate(x)])/np.sum(x)
Minimun value of intra class variance 126
Maximum value of inter class variance 126
```



Q. 3 Images are superimposed by looping the RGB plane and assigning the value of text image into background image corresponding to each plane.



Q. 4 Connected components analysis

```
C:\Users\ajeet\Desktop\Digital Image Processing\Signal Processir
  x3 = np.sum([(j-x2)**2*t for j,t in enumerate(x)])/np.sum(x)
Number of shapes: 30
Number of circles: 19
PS C:\Users\ajeet\Desktop\Digital Image Processing\Signal Proces
```

Note : In a given image there is a single pixel having pixel value 255 and algorithm is counting that as a single object hence the count is 30.

[771769, 526, 532, 522, 528, 13333, 524, 13230, 13230, 518, 522, 524, 14031, 17462, 527, 523, 13230, 536, 531, 13988, 14124, 522, 529, 526, 1, 13230, 13980, 525, 524, 522, 531]

This thing can be removed simply by applying a condition (if pixel count <50 : ignore that) .

```
number_of_shapes=k-1
counter = [0 for i in range(k)]
for i in range(row):
    for j in range(col):
        counter[int(R[i,j])] = counter[int(R[i,j])] + 1
circle_count = 0
number_of_shapes1 = 0
for k in counter:
    if k>50:
        number_of_shapes1 = number_of_shapes1+ 1
    if k>500 and k<550:
        circle_count = circle_count+1
number_of_shapes1 = number_of_shapes1 -1
return number_of_shapes, circle_count, counter, number_of_shapes1
```

And now output is

```
  scalars
    x3 = np.sum([(j-x2)**2*t for j,t in enumerate(x)])/np.sum(x)
  Number of shapes: 29
  Number of circles: 19
  PS C:\Users\ajeet\Desktop\Digital Image Processing\Signal Processing\Assignment_1>
```