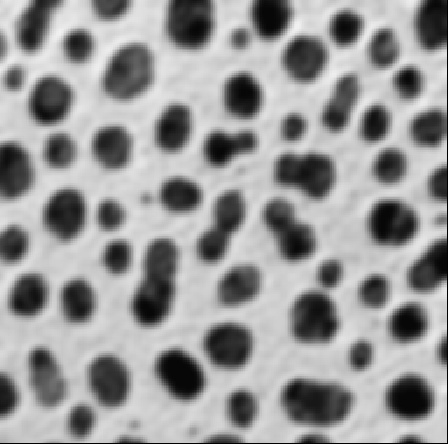
Assignment-1

1.a) Nearest\_neighbor :

\* For the given resizing factors, an empty grid is generated with dimensions obtained by multiplying the number of rows and columns of the original image with resize factor and the values are assigned using interpolation.

\* Then, grid is shrinked to size of the original image.

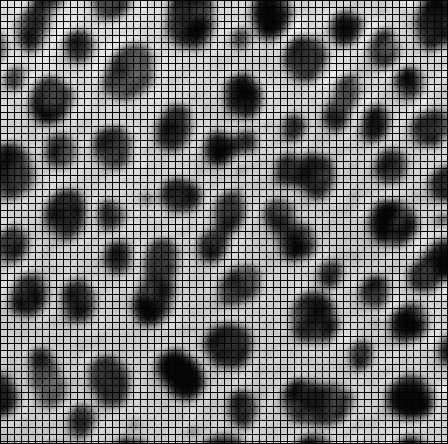
\* For each pixel value in the grid, a comparison to the input image is made, and the value that is nearest in the input image is found and assigned to the pixel in the output grid.



1.b) Bilinear Interpolation:

\* First, the linear interpolation is applied to all the neighboring pixels for all possible pairs except the ones that are diagonal to each other.

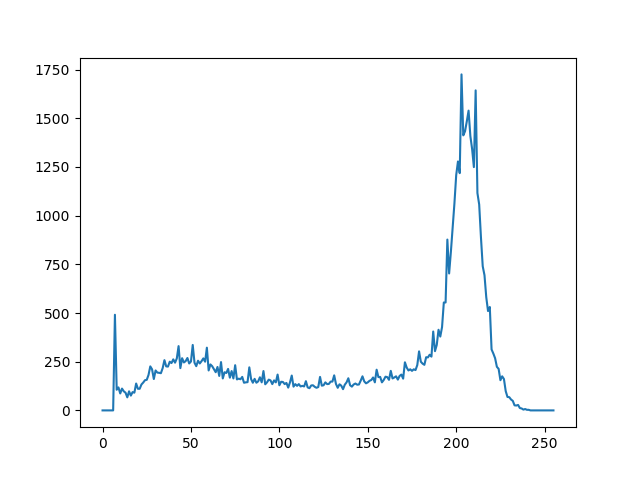
\* Then, the weighted average of the linearly interpolated values are calculated and assigned to the pixels in the output image.



2.a)

Computing histogram:

\* After reading the image, no. of values corresponding to each intensity value are plotted on y-axis to obtain a histogram.



Computing Threshold:

\* Considering a random value to be the initial threshold value, assuming the distribution to be bimodal, means of the partitions: from minimum value of the intensity to threshold and from threshold to maximum intensity value, are calculated.

\* A new threshold value is calculated using the average of the means calculated.

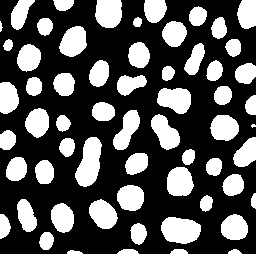
\* This process is continued until the value of the calculated threshold value and average of the means converge.

\* The obtained threshold value is “130”

Computing Binary Image:

\* Based on the threshold value calculated, assuming background to be “255” and foreground to be “0”, all the values “less” than threshold are assigned to the value “255” and all the values “greater” than threshold are assigned “0”.

\* The output binarized image is stored in the output folder.



Blob Coloring:

\* For the obtained binarized image, a “region color” array “R” is defined, in which the region number corresponding to each pixel is stored.

\* Initially all the elements of “R” are initialized to “0” and a “count\_val” variable which gives the number of regions is taken and initialized to “1”

\* Considering the pixels that are at left,top-left, above and top-right to the given pixel, the regions are assigned to the pixels as follows:

\* if I[i, j] == 255 and I[i, j - 1] == 0 and I[i - 1, j - 1] == 0 and I[i - 1, j] == 0 and I[i - 1, j + 1] == 0:

set R[i, j] = count\_val

count\_val = count\_val + 1

if I[i, j] == 255 and I[i, j - 1] == 0 and I[i - 1, j - 1] == 0 and I[i - 1, j] == 0 and I[ i- 1, j + 1] == 255:

set R[i, j] = R[i - 1, j + 1]

if I[i, j] == 255 and I[i, j - 1] == 0 and I[i - 1, j - 1] == 0 and I[i - 1, j] == 255 and I[i - 1, j + 1] == 0:

set R[i, j] = R[i - 1, j]

if I[i, j] == 255 and I[i, j - 1] == 0 and I[i - 1, j - 1] == 255 and I[i - 1, j] == 0 and I[i - 1, j + 1] == 0:

set R[i, j] = R[i - 1, j - 1]

if I[i, j] == 255 and I[i, j - 1] == 255 and I[i - 1, j - 1] == 0 and I[i - 1, j] == 0 and I[i - 1, j + 1] == 0:

set R[i, j] = R[i, j - 1]

\* A region array is printed which gives the indices of the pixels (i,j) that is the location of the pixels belonging to the regions. The location of the pixels present in various regions are as follows:

Regions: {1: [[0, 19], [0, 20], [0, 21], [0, 22], [0, 23], [0, 24], [0, 25], [0, 26], [0, 27], [0, 28], [0, 29], [0, 30], [0, 31], [0, 32], [0, 33], [0, 34], [1, 18], [1, 19], [1, 20], [1, 21], [1, 22], [1, 23], [1, 24], [1, 25], [1, 26], [1, 27], [1, 28], [1, 29], [1, 30], [1, 31], [1, 32], [1, 33], [1, 34], [2, 18], [2, 19], [2, 20], [2, 21], [2, 22], [2, 23], [2, 24], [2, 25], [2, 26], [2, 27], [2, 28], [2, 29], [2, 30], [2, 31], [2, 32], [2, 33], [3, 17], [3, 18], [3, 19], [3, 20], [3, 21], [3, 22], [3, 23], [3, 24], [3, 25], [3, 26], [3, 27], [3, 28], [3, 29], [3, 30], [3, 31], [3, 32], [4, 17], [4, 18], [4, 19], [4, 20], [4, 21], [4, 22], [4, 23], [4, 24], [4, 25], [4, 26], [4, 27], [4, 28], [4, 29], [4, 30], [4, 31], [5, 16], [5, 17], [5, 18], [5, 19], [5, 20], [5, 21], [5, 22], [5, 23], [5, 24], [5, 25], [5, 26], [5, 27], [5, 28], [5, 29], [5, 30], [6, 15], [6, 16], [6, 17], [6, 18], [6, 19], [6, 20], [6, 21], [6, 22], [6, 23], [6, 24], [6, 25], [6, 26], [6, 27], [6, 28], [6, 29], [7, 14], [7, 15], [7, 16], [7, 17], [7, 18], [7, 19], [7, 20], [7, 21], [7, 22], [7, 23], [7, 24], [7, 25], [7, 26], [7, 27], [7, 28], [8, 14], [8, 15], [8, 16], [8, 17], [8, 18], [8, 19], [8, 20], [8, 21], [8, 22], [8, 23], [8, 24], [8, 25], [8, 26], [8, 27], [8, 28], [9, 13], [9, 14], [9, 15], [9, 16], [9, 17], [9, 18], [9, 19], [9, 20], [9, 21], [9, 22], [9, 23], [9, 24], [9, 25], [9, 26], [9, 27], [9, 28], [10, 13], [10, 14], [10, 15], [10, 16], [10, 17], [10, 18], [10, 19], [10, 20], [10, 21], [10, 22], [10, 23], [10, 24], [10, 25], [10, 26], [10, 27], [11, 12], [11, 13], [11, 14], [11, 15], [11, 16], [11, 17], [11, 18], [11, 19], [11, 20], [11, 21], [11, 22], [11, 23], [11, 24], [11, 25], [11, 26], [11, 27], [12, 12], [12, 13], [12, 14], [12, 15], [12, 16], [12, 17], [12, 18], [12, 19], [12, 20], [12, 21], [12, 22], [12, 23], [12, 24], [12, 25], [12, 26], [12, 27], [13, 11], [13, 12], [13, 13], [13, 14], [13, 15], [13, 16], [13, 17], [13, 18], [13, 19], [13, 20], [13, 21], [13, 22], [13, 23], [13, 24], [13, 25], [13, 26], [13, 27], [14, 11], [14, 12], [14, 13], [14, 14], [14, 15], [14, 16], [14, 17], [14, 18], [14, 19], [14, 20], [14, 21], [14, 22], [14, 23], [14, 24], [14, 25], [14, 26], [15, 11], [15, 12], [15, 13], [15, 14], [15, 15], [15, 16], [15, 17], [15, 18], [15, 19], [15, 20], [15, 21], [15, 22], [15, 23], [15, 24], [15, 25], [15, 26], [16, 11], [16, 12], [16, 13], [16, 14], [16, 15], [16, 16], [16, 17], [16, 18], [16, 19], [16, 20], [16, 21], [16, 22], [16, 23], [16, 24], [16, 25], [16, 26], [17, 11], [17, 12], [17, 13], [17, 14], [17, 15], [17, 16], [17, 17], [17, 18], [17, 19], [17, 20], [17, 21], [17, 22], [17, 23], [17, 24], [17, 25], [17, 26], [18, 11], [18, 12], [18, 13], [18, 14], [18, 15], [18, 16], [18, 17], [18, 18], [18, 19], [18, 20], [18, 21], [18, 22], [18, 23], [18, 24], [18, 25], [19, 11], [19, 12], [19, 13], [19, 14], [19, 15], [19, 16], [19, 17], [19, 18], [19, 19], [19, 20], [19, 21], [19, 22], [19, 23], [19, 24], [19, 25], [20, 11], [20, 12], [20, 13], [20, 14], [20, 15], [20, 16], [20, 17], [20, 18], [20, 19], [20, 20], [20, 21], [20, 22], [20, 23], [20, 24], [21, 11], [21, 12], [21, 13], [21, 14], [21, 15], [21, 16], [21, 17], [21, 18], [21, 19], [21, 20], [21, 21], [21, 22], [21, 23], [21, 24], [22, 11], [22, 12], [22, 13], [22, 14], [22, 15], [22, 16], [22, 17], [22, 18], [22, 19], [22, 20], [22, 21], [22, 22], [22, 23], [22, 24], [23, 11], [23, 12], [23, 13], [23, 14], [23, 15], [23, 16], [23, 17], [23, 18], [23, 19], [23, 20], [23, 21], [23, 22], [23, 23], [24, 11], [24, 12], [24, 13], [24, 14], [24, 15], [24, 16], [24, 17], [24, 18], [24, 19], [24, 20], [24, 21], [24, 22], [24, 23], [25, 12], [25, 13], [25, 14], [25, 15], [25, 16], [25, 17], [25, 18], [25, 19], [25, 20], [25, 21], [25, 22], [26, 12], [26, 13], [26, 14], [26, 15], [26, 16], [26, 17], [26, 18], [26, 19], [26, 20], [26, 21], [26, 22], [27, 13], [27, 14], [27, 15], [27, 16], [27, 17], [27, 18], [27, 19], [27, 20], [27, 21], [28, 14], [28, 15], [28, 16], [28, 17], [28, 18], [28, 19], [28, 20], [29, 16], [29, 17], [29, 18]]

2: [[0, 53], [0, 54], [0, 55], [0, 56], [0, 57], [0, 58], [0, 59], [0, 60], [0, 61], [0, 62], [1, 54], [1, 55], [1, 56], [1, 57], [1, 58], [1, 59], [1, 60], [2, 54], [2, 55], [2, 56], [2, 57], [2, 58], [3, 54], [3, 55], [3, 56], [4, 54]],

3: [[0, 97], [0, 98], [0, 99], [0, 100], [0, 101], [0, 102], [0, 103], [0, 104], [0, 105], [0, 106], [0, 107], [0, 108], [0, 109], [0, 110], [0, 111], [0, 112], [1, 97], [1, 98], [1, 99], [1, 100], [1, 101], [1, 102], [1, 103], [1, 104], [1, 105], [1, 106], [1, 107], [1, 108], [1, 109], [1, 110], [2, 97], [2, 98], [2, 99], [2, 100], [2, 101], [2, 102], [2, 103], [2, 104], [2, 105], [2, 106], [2, 107], [2, 108], [3, 97], [3, 98], [3, 99], [3, 100], [3, 101], [3, 102], [3, 103], [3, 104], [3, 105], [3, 106], [4, 97], [4, 98], [4, 99], [4, 100], [4, 101], [4, 102], [4, 103], [4, 104], [5, 96], [5, 97], [5, 98], [5, 99], [5, 100], [5, 101], [5, 102], [6, 96], [6, 97], [6, 98], [6, 99], [6, 100], [7, 96], [7, 97], [7, 98], [8, 96]]

.

.

.

131: [[250, 70], [250, 71], [250, 72], [250, 73], [251, 68], [251, 69], [251, 70], [251, 71], [251, 72], [251, 73], [251, 74], [251, 75], [251, 76], [251, 77], [251, 78], [251, 79], [252, 67], [252, 68], [252, 69], [252, 70], [252, 71], [252, 72], [252, 73], [252, 74], [252, 75], [252, 76], [252, 77], [252, 78], [252, 79], [252, 80], [252, 81]], 132: [], 133: [], 134: [], 135: [], 136: [], 137: []}

Region Analysis:

All the regions which number of pixels less than 15 are ignored. The statistics for the remaining regions : centroid and area are calculated and are as follows :

The format specified for the information above is :

[Region\_Number,[centroid(x,y)], Area]

1: [[13, 20], 423]

2: [[1, 56], 26]

3: [[2, 101], 76]

4: [[9, 154], 433]

5: [[15, 197], 276]

6: [[21, 137], 71]

7: [[27, 218], 260]

8: [[26, 44], 222]

9: [[33, 173], 489]

10: [[41, 73], 635]

.

.

50: [[227, 234], 555]

51: [[229, 180], 845]

52: [[233, 138], 273]

53: [[240, 46], 208]

54: [[250, 127], 65]

55: [[250, 178], 56]

56: [[250, 234], 36]

57: [[251, 73], 31]

