HW 2 Basic Image Manipulation

1. Binarize lena.bmp with threshold 128

Source code

Please refer to the file “main.cpp” within the same folder as this report document.
(from line#30 to line#40)

Result

(the resulted image is saved as lena_binarized.bmp within the same folder as well)

2. Draw the histogram

Source code

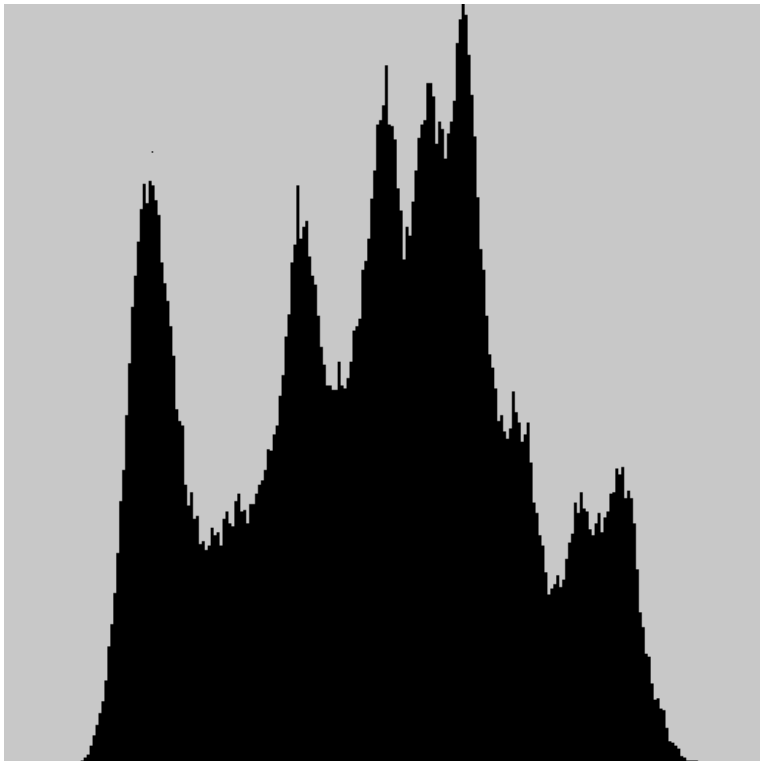
Please refer to the file “main.cpp” within the same folder as this report document.
(from line#42 to line#70)

Steps That I Took

Step #1: Count the intensity (loop through all pixels in lena.bmp)
Step #2: Create a “Mat” variable for drawing image
Step #3: Draw the histogram with code (hardcored)
Step #4: Display and Save the image (using imshow() and imwrite() functions)

Result

(the resulted image is saved as lena_histogram.bmp within the same folder as well)



3. Find Connected Component

Source code

Please refer to the file “main.cpp” within the same folder as this report document.
(from line#72 to line#295)

About

First of all, I regarded “128~255” as the part we have interests in.

The algorithm I used is “The Classical Algorithm” (the 2nd one of the three).

From line#72 to line#179
Loop through each pixel and record the equivalence in global equivalence table.

From line#197 to line#208
Restore every pixel to the correct connected component through global equivalence table.

From line#222 to line#248
Calculate how many pixels in each connected component, and the Left, Right, Up, Bottom margin of them.

From line#266 to line#295
Draw the bounding boxes we found.
(I use the binarized image as background, but in order to see the bounding boxes more clearly, I change all the pixels with 0 intensity to 100. And the bounding boxes are displayed with intensity 0.)

Result

(the resulted image is saved as lena_connected_component.bmp within the same folder as well)

There are five bounding boxes. (lena_connected_component.bmp)