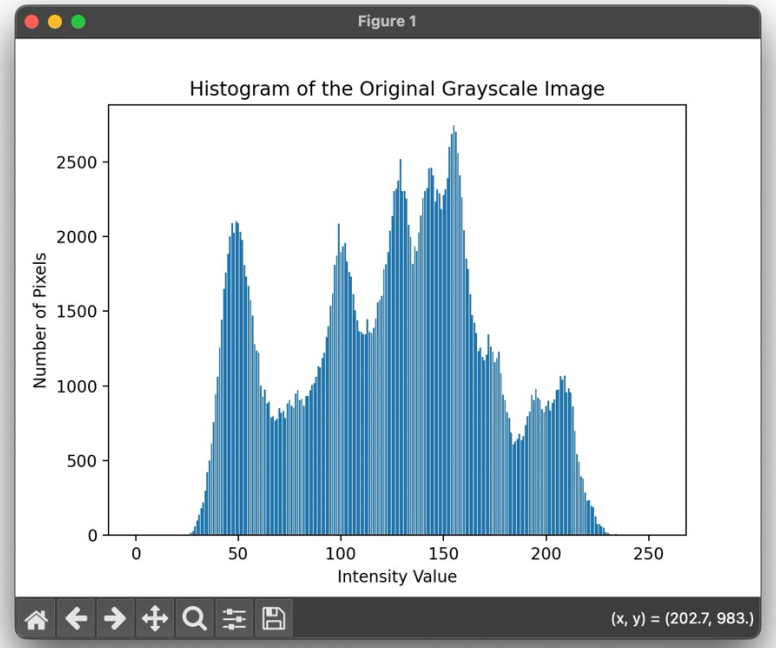
R13922136 周洛君

Computer Vision Homework 2: Report

The code in this homework contains three parts, the first is to binarize the image, the second is to generate the histogram, and the last is to find the connected components, mark with boxes and centroids.

In the first part, I do similar things as the last homework, which is to first create an image having the same shape as the lena.bmp image, but initialized with 0. Then all we have to do is to set the pixels with corresponding pixel in the original image having intensity value greater or equal to 128, to 255. The resulting image is as follows:

In the second part, to create the histogram, I first create an array having size 256 to record the number of pixels corresponding to each intensity value. After that, we just loop through all the pixels of the image and add one record to its value in the array. To plot the histogram, I use plt.bar in matplotlib. The histogram is as shown below:



For the third part, to find the connected components, we first have to assign an unique label to each pixel having intensity value greater than zero. This is done simply by initializing an image with same shape, having all pixels with value zero, then a variable “uniq_label” is used to mark the current label number that should be assigned to the next nonzero pixel.

After the initialization above, we then proceed to do iterations of top-down followed by bottom-up passes to update the label values. Actually, the top-down pass is almost the same as the bottom-up pass, except that for the latter, we start from the bottom right corner and goes to the top left corner. This is done by setting the loop to start from height / width – 1, to -1, with each step of -1.

No matter it’s top-down or bottom-up, we split all situations into 9 cases, the previous 4 is to deal with the four pixels on the four corners, then the next 4 cases aimed to apply on those on the edges (except pixels contained by the first 4 cases), so in the first 8 cases we covered all the pixels along the edges of the image, therefore all the inner pixels are classified to the last case.

In each case, things are quite similar, I chose to find 4-connected components, so we just first loop through each pixel and check if its value is greater than zero, then check if there exists any of its four neighbors (or if on the edges then less than 4) having label number smaller than this current pixel or not. If so, we update the variable “min_label” by its smallest neighbor. Next, since we finish our update if none of the pixels are changed in an iteration, we use a flag named “change” to record if there’s any change during the iteration. Thus, after checking the neighbor pixels and possibly changing “min_label”, we then check if “min_label” is still the same as the current pixel label, if it’s not, the flag is marked as True, and all the neighbors are updated by “min_label”.

Next, in order to apply a 500 pixels threshold, we first calculate the number of each label, this is done similarly as the counting done in the histogram part. With the result, we then store the labels having counts more greater or equal to 500 to another array named “target_label_arr”. The next step is to find the minimum and maximum x, y values from the pixels having the same label for each label recorded in “target_label_arr”. After this is done, we can plot the location of each rectangle. Note that before this step, we first create a modified copy of the lena image with BGR format, so that we can draw colored rectangles on it (also the centroids). We finally draw the rectangles by the cv2.rectangle function.

The last thing to do is to plot the centroids of each box. To find the location of each centroid, we calculate the mean value of the x-axis and y-axis from the pixels in the same rectangle. Other things are similar to what we have done in recording the rectangle locations. I use cv2.circle to plot the centroids.

There are pretty more details in this part, like some things must be modified according to the OpenCV convention, or the type of function parameters……, all of these are mentioned in the comments.

The following image is my result:


Lena with Rectangles and Centroids