Homework Report for Computer Vision

Yu Xiang, Luo

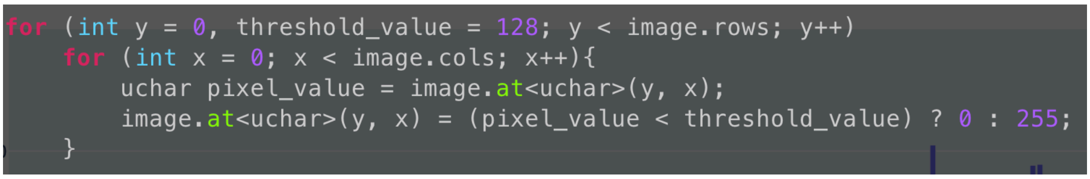
September 22, 2023

Minor parts of my code take reference from this github

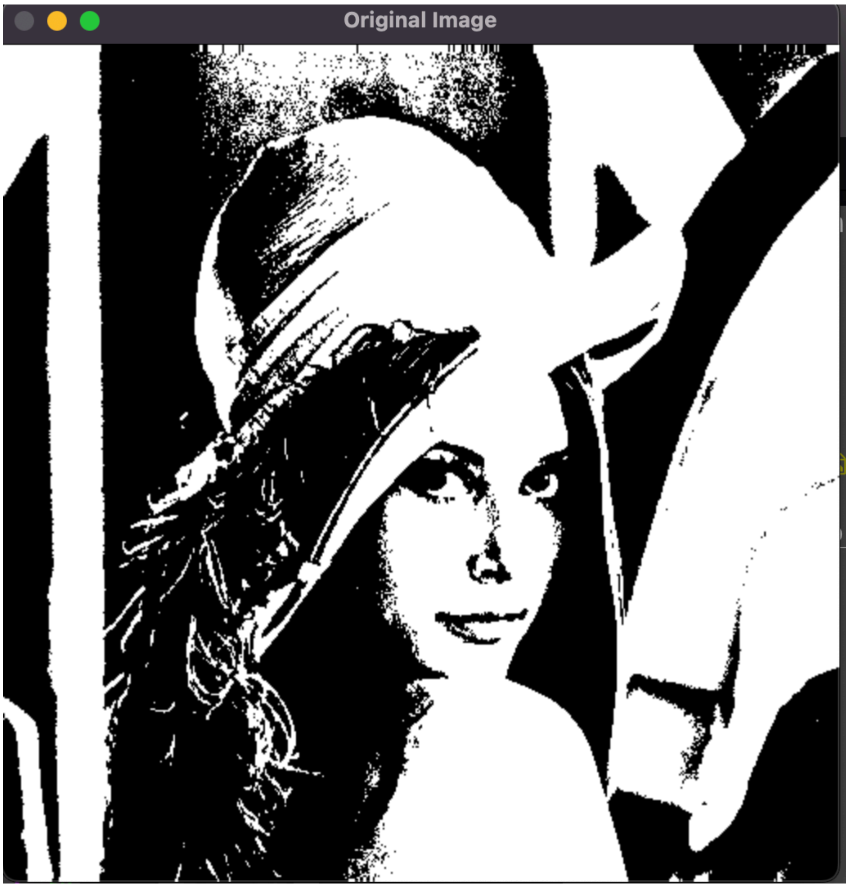
You can check this github for more information

A Binary Image

A naive **for** loop can solve it.

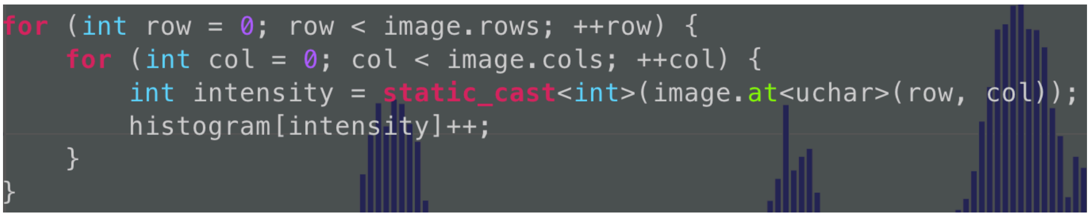


```
for (int y = 0, threshold_value = 128; y < image.rows; y++)
    for (int x = 0; x < image.cols; x++){
        uchar pixel_value = image.at<uchar>(y, x);
        image.at<uchar>(y, x) = (pixel_value < threshold_value) ? 0 : 255;
    }
```

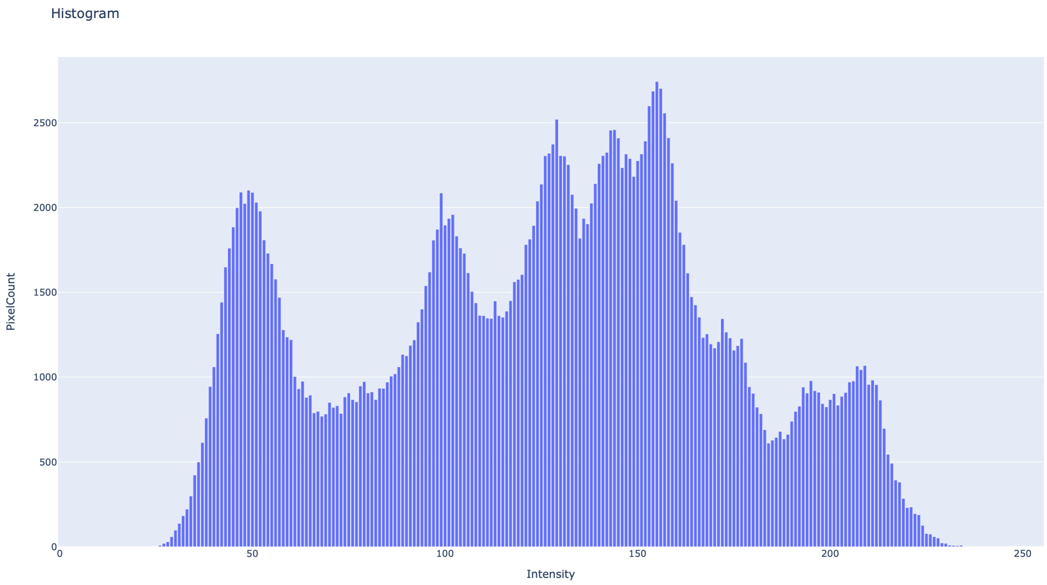


Histogram

Find the intensity at each pixel, then store it to the array **histogram**. Store the array to an csv file and use plotly to draw the histogram.

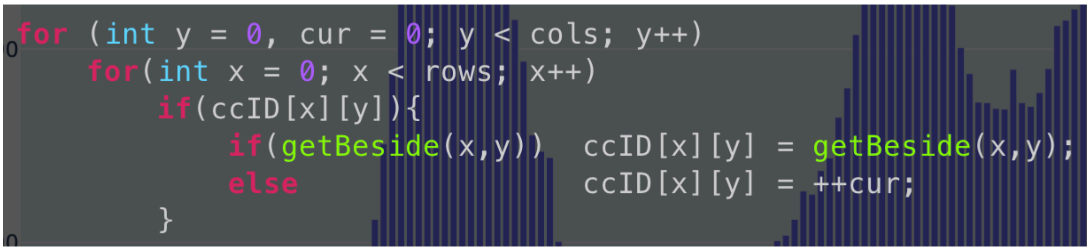


```
for (int row = 0; row < image.rows; ++row) {
    for (int col = 0; col < image.cols; ++col) {
        int intensity = static_cast<int>(image.at<uchar>(row, col));
        histogram[intensity]++;
    }
}
```



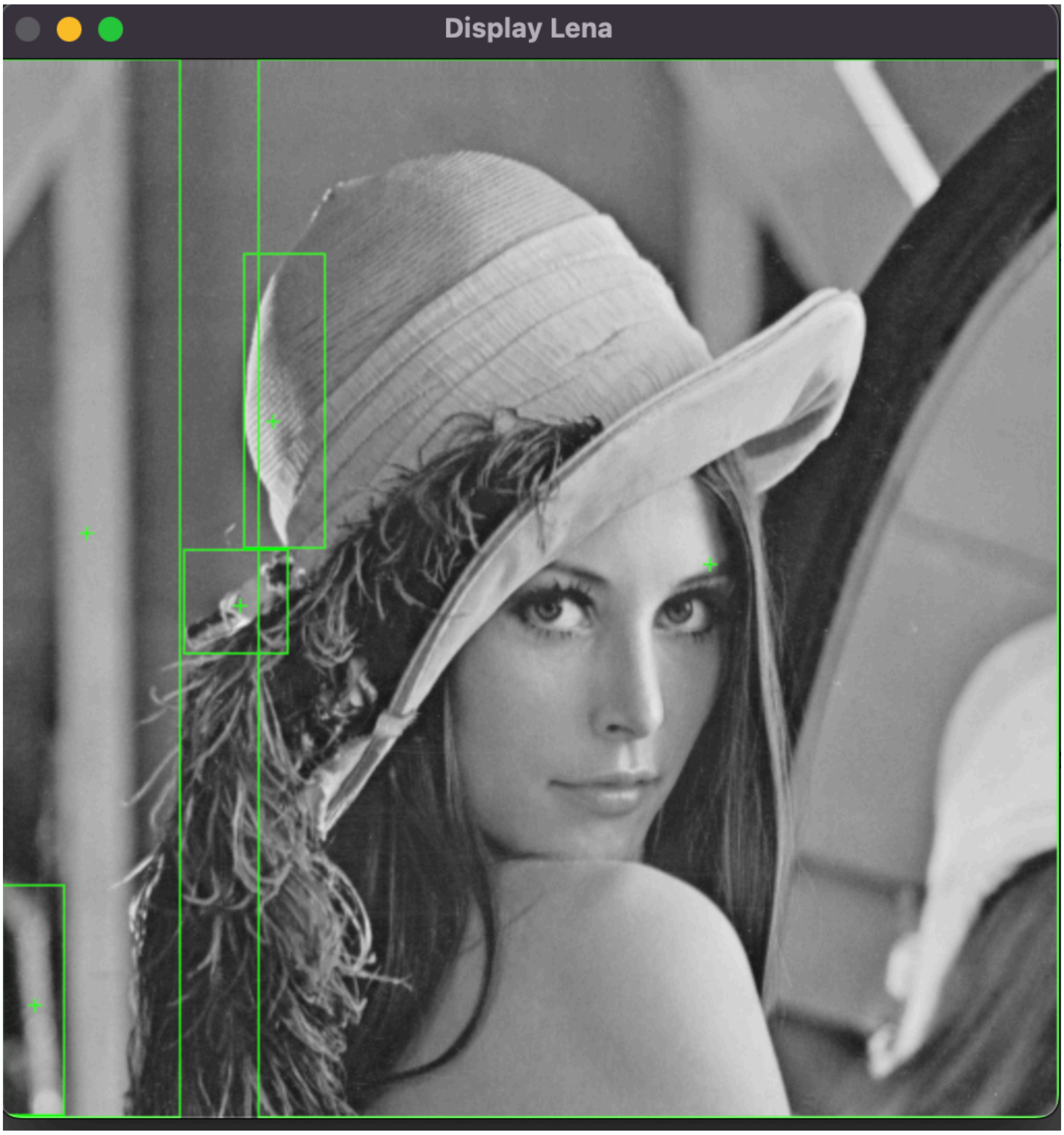
Connected Conponents – 4-connected

Since one **cc**(connected conponent) must connect to its neighbor, so my implementation is based on this property. I'm using an array ccID to store different **cc**'s ID.



```
for (int y = 0, cur = 0; y < cols; y++)
    for(int x = 0; x < rows; x++)
        if(ccID[x][y]){
            if(getBeside(x,y))  ccID[x][y] = getBeside(x,y);
            else                ccID[x][y] = ++cur;
        }
```

If the pixel beside the current pixel has already been assigned a ID, then current pixel use that as its own ID, else assign a new ID to it. In this way, we achieve the concept of **cc**: Pixels that connect to each is one conponents(share same ID).



Others

For details:

1. How to draw a graph: PLZ take reference to graph.py

2. How to store histogram as csv file: PLZ take reference to histogram.cpp

3. How to draw center and line at **cc**: Use ccID to find the leftmost, rightmost, upmost, downmost index and label them as min..., then use the function **`drawline`**, **`drawcenter`** to mark ccID as -1, then color them in the end.