Computer Vision I: Homework 2

Department: CSIE   Student ID: d11922016   Name: Jia-Wei, Liao

September 27, 2022

Write a program to generate:

(a) a binary image (threshold at 128)

(b) a histogram

(c) connected components (regions with + at centroid, bounding box)

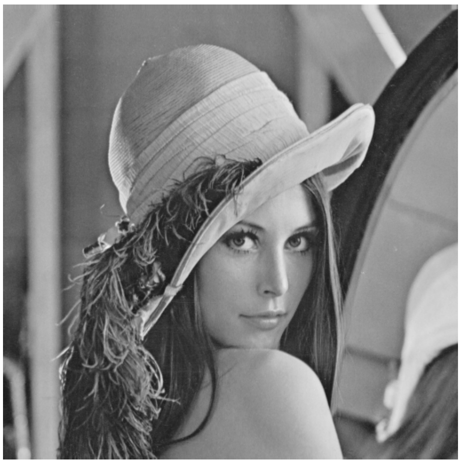


Figure 1: raw image

1 Binary Image

Let I be the intensity of image. The binary image I' represent as

$$I'(i,j) = \begin{cases} 255, & \text{if } I(i,j) > 128 \\ 0, & \text{if } I(i,j) \leq 128 \end{cases}$$

```
for i = 1:m
    for j = 1:n
        if img(i, j) >= 128
            binary_img(i, j) = 255;
        else
            binary_img(i, j) = 0;
        end
    end
end
```

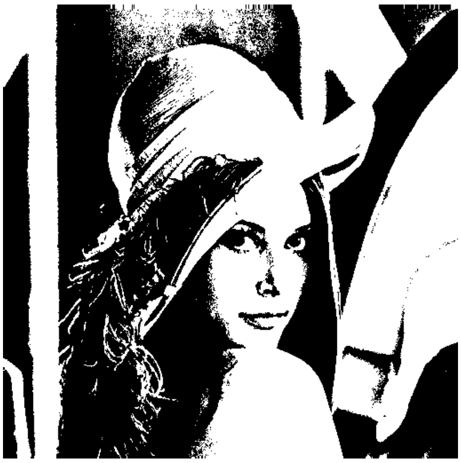


Figure 2: binary image

2 Histogram of Image Intensity

Let Ω be the domain of the image and $I(x, y)$ be the intensity of image. The value of histogram can compute as the following:

$$h(i) = \int_{(x,y)\in\Omega} \mathbb{1}_{\{I(x,y)=i\}} dxdy$$

```
for g = 1:255
    for i = 1:m
        for j = 1:n
            if img(i, j) == g
                img_hist(1, g) = img_hist(1, g) + 1;
            end
        end
    end
end
```

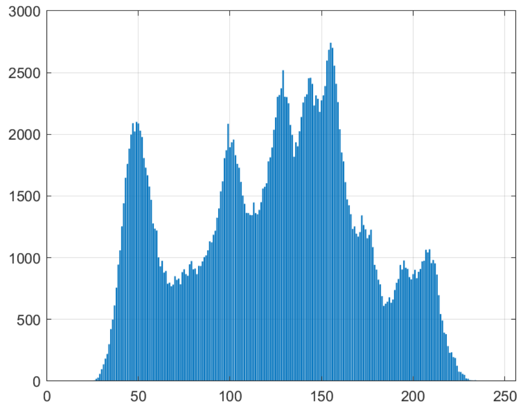


Figure 3: image histogram

3 Connected Components Image

1. We implement an 8-connected components algorithm, which needs to check 4 adjacent pixels, upper left, upper, upper right, and left, respectively.

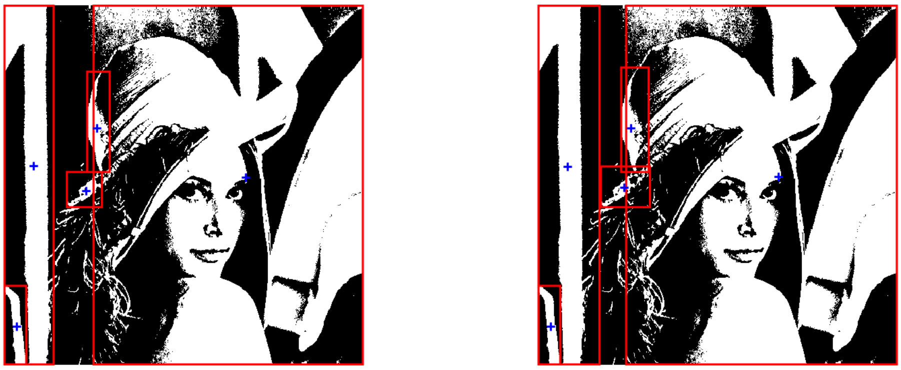
2. The centroid (x_c, y_c) of the component C is represent as

$$x_c = \frac{1}{Area(C)} \int_{(x,y)\in C} x dx$$
$$y_c = \frac{1}{Area(C)} \int_{(x,y)\in C} y dy$$

```
% First pass
for i = 1:m
    for j = 1:n
        if binary_img(i, j) > 0
            neighbor_labels = getNeightborLab(i, j, labels, n_connected);
            if isempty(neighbor_labels)
                labels(i, j) = next_label;
                linked{next_label} = [next_label];
                next_label = next_label + 1;
            else
                labels(i, j) = min(neighbor_labels);
                for k = neighbor_labels
                    for l = neighbor_labels
                        if k ~= l
                            linked{k} = union(linked{k}, linked{l});
                        end
                    end
                end
            end
        end
    end
end

for i = 1:length(linked)
    for j = linked{i}
        linked{i} = union(linked{i}, linked{j});
    end
end

% Second pass
for i = 1:m
    for j = 1:n
        if labels(i, j) > 0
            labels(i, j) = min(linked{labels(i, j)});
        end
    end
end
```



(a) 4 connected components

(b) 8 connected components

Figure 4: connected components image

Summary.

This homework aims to find the connected components of the binary image. We use MATLAB to implement the algorithm. Since prohibiting vectorized computation, we use for loop to deal with pixels individually. Finally, we show the 5 disjoint connected components which contained above 500 pixels and plotted the bounding boxes and their centroid point.

4 Reference

1. `https://en.wikipedia.org/wiki/Connected-component_labeling`