電腦視覺 Homework 2 – Basic Image Manipulation

資工碩一 r08922a04 林承德

Language : python 3.7
Library : numpy, PIL, matplotlib
Execution way: python3 hw2.py --lena lena.bmp

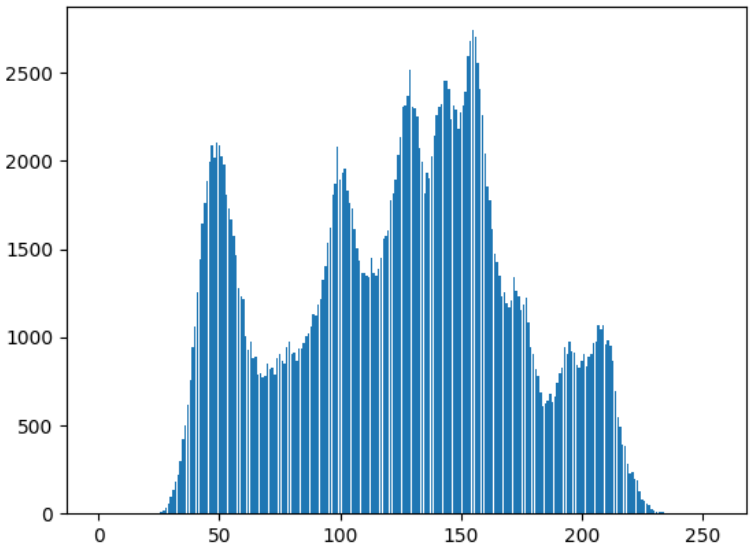
Task:

(a) a binary image (threshold at 128)

```python
class Question1():
    def run(self, threshold, path):
        image = Image.open(path)
        h, w = image.size
        for i in range(h):
            for j in range(w):
                value = image.getpixel((i, j))
                value = 255 if value >= threshold else 0
                image.putpixel((i,j), value)
        image.save('binary.bmp')
```

I go through all pixels with 2 layer- for-loop, and if any pixel's value is greater or equal than the threshold, I will force it to be 255 and the pixel which it's value is less than threshold, I will force it to be 0.

(b) a histogram



```python
class Question2():
    def run(self, path, dim):
        his = np.zeros(dim)
        image = Image.open(path)
        h, w = image.size
        for i in range(h):
            for j in range(w):
                val = image.getpixel((i, j))
                his[val] += 1
        np.savetxt("histogram.csv", his, delimiter=",")
        plt.bar(range(dim), his)
        plt.savefig('histogram.png')
        plt.show()
```

I use one dimension numpy array (0~255)to store the number of how many pixel in this image have this value(0~255), then output as a csv file with np.savetxt. Finally draw a histogram by matplotlib.

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

```python
class Question3():
    def run(self, path, threshold):
        def preprocess(h, w, bin_image):
            vis = np.zeros((h, w))
            label = np.zeros((h, w))
            region_id = 1
            n_labels = np.zeros(h * w)
            for row in range(h):
                for col in range(w):
                    if bin_image.getpixel((row, col)) == 0:
                        vis[row, col] = 1
                    elif vis[row, col] == 0:
                        stack = []
                        stack.append((row, col))
                        while len(stack) > 0:
                            r, c = stack.pop()
                            if vis[r, c] == 1:
                                continue
                            vis[r, c] = 1
                            label[r, c] = region_id
                            n_labels[region_id] += 1
                            for y in [r - 1, r, r + 1]:
                                for x in [c - 1, c, c + 1]:
                                    if (0 <= y < h) and (0 <
                                        if (bin_image.getpix
                                            stack.append((y,
                        region_id += 1
            return label, n_labels
```

```python
        def rectangle(label, n_labels, thres,  h, w):
            recs = []
            for region, n in enumerate(n_labels):
                if (n >= thres):
                    left = w
                    right = 0
                    top = h
                    bot = 0
                    for y in range(w):
                        for x in range(h):
                            if (label[y, x] == region):
                                if (y < left):
                                    left = y
                                if (y > right):
                                    right = y
                                if (x < top):
                                    top = x
                                if (x > bot):
                                    bot = x
                    recs.append((left, right, top, bot))
            return recs
```

```python
    def draws(recs, rec_img):
        while recs:
            left, right, top, bot = recs.pop()
            draw = ImageDraw.Draw(rec_img)
            draw.rectangle(((left, top), (right, bot)), outline = 'yellow')
            centroid_x = (left + right) / 2
            centroid_y = (top + bot) / 2
            draw.line(((centroid_x - 10, centroid_y), (centroid_x + 10, centroid_y)), fill = 'green', width = 3)
            draw.line(((centroid_x, centroid_y - 10), (centroid_x, centroid_y + 10)), fill = 'green', width = 3)
        _rec_img.save('connect_components.bmp')
```

```python
hreshold = 500
bin_img = Image.open('binary.bmp')
h, w = bin_img.size
label,n_labels = preprocess(h, w, bin_img)
recs = rectangle(label, n_labels, threshold, w, h)
rec_img = Image.open('binary.bmp').convert('RGB')

draws(recs, rec_img)
```

Briefly, in first step, I find out the connect component with 8連通, then using these information to find out the edge of the rectangles, finally use DrawImage to draw these rectangle in yellow and the note the centroid of every rectangle with green cross.