I. Problem Statement

- Part1. Write a program to generate the followings.
  (a) a binary image (threshold at 128)
  (b) a histogram
  (c) connected components(regions with + at centroid, bounding box)

II. Programming Tools

- Programming language: Python 3.7.7
- Library: Numpy 1.19.0, OpenCV 3.4.2, matplotlib 3.2.2

III. Problem-Solving Process

a. a binary image (threshold at 128)

```
if choice == 0: # a binary image
    ret_img = np.where(img > 128, 255, 0)
    cv2.imwrite('./output/1.jpg', ret_img)
```

b. a histogram

```
elif choice == 1: # a histogram
    statistic = np.zeros(256)
    r, c = img.shape
    for i in range(r):
        for j in range(c):
            statistic[img[i,j]] += 1

    plt.style.use('seaborn-white')
    plt.bar(range(256) ,statistic)
    plt.xlabel('pixel value')
    plt.ylabel('number')
    plt.savefig('./output/2.jpg')
```

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

```
elif choice == 2: # connected components
    row, col = img.shape
    ret_img = np.where(img > 128, 255, 0)
    ret_map = np.zeros_like(ret_img)
    label = 1
    # initialization of each 1-pixel to a unique label
    for r in range(row):
        for c in range(col):
            if ret_img[r,c] == 255:
                ret_map[r,c] = label
                label += 1
```

資工所 r08922123
Author: 王韻豪

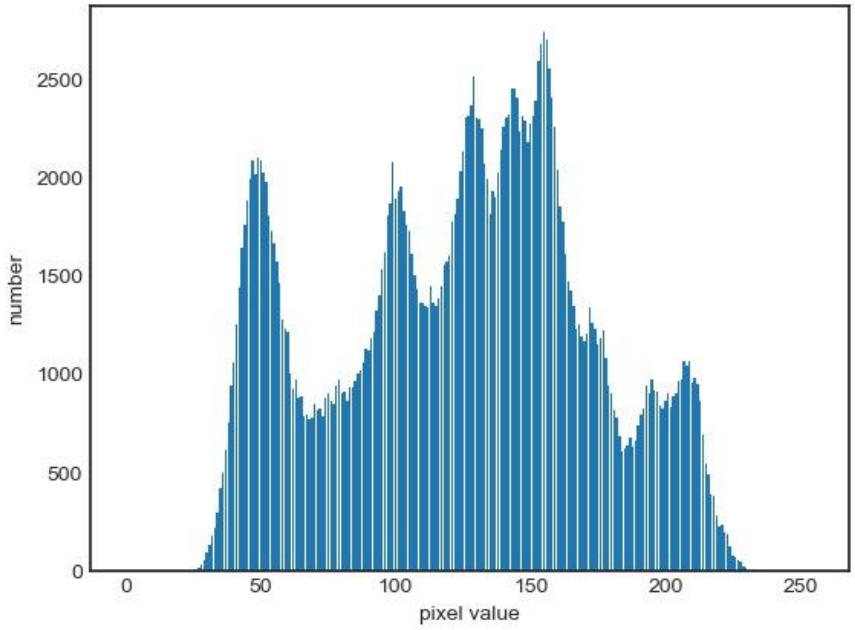
```python
# iteration of top-down followed by bottom-up passes
change = True
while change:
    change = False
    for r in range(row):
        for c in range(col):
            if ret_map[r,c] != 0:
                for i in range(-1,2,1):
                    if r+i >= row or r+i < 0:
                        continue
                    for j in range(-1,2,1):
                        if c+j >= col or c+j < 0:
                            continue
                        if ret_map[r+i, c+j] != 0:
                            if ret_map[r,c] > ret_map[r+i, c+j]:
                                ret_map[r,c] = ret_map[r+i, c+j]
                                change = True
    for r in range(row-1,-1,-1):
        for c in range(col-1,-1,-1):
            if ret_map[r,c] != 0:
                for i in range(-1,2,1):
                    if r+i >= row or r+i < 0:
                        continue
                    for j in range(-1,2,1):
                        if c+j >= col or c+j < 0:
                            continue
                        if ret_map[r+i, c+j] != 0:
                            if ret_map[r,c] > ret_map[r+i, c+j]:
                                ret_map[r,c] = ret_map[r+i, c+j]
                                change = True
```

```python
centroid_map = dict()
box_map = dict()
for r in range(row):
    for c in range(col):
        if ret_map[r,c] != 0:
            if ret_map[r,c] not in centroid_map:
                centroid_map[ret_map[r,c]] = [(r,c)]
                box_map[ret_map[r,c]] = {'u':r,'d':r,'l':c,'r':c}
            else:
                centroid_map[ret_map[r,c]].append((r,c))
                box_map[ret_map[r,c]]['u'] = min(box_map[ret_map[r,c]]['u'], r)
                box_map[ret_map[r,c]]['d'] = max(box_map[ret_map[r,c]]['d'], r)
                box_map[ret_map[r,c]]['l'] = min(box_map[ret_map[r,c]]['l'], c)
                box_map[ret_map[r,c]]['r'] = max(box_map[ret_map[r,c]]['r'], c)

ret_img = cv2.cvtColor(ret_img.astype(np.uint8), cv2.COLOR_GRAY2BGR)
for key in centroid_map:
    if len(centroid_map[key]) > 500:
        cv2.rectangle(ret_img, (box_map[key]['l'],box_map[key]['u']), (box_map[key]['r'],box_map[key]['d']), (0,255,0), 4)
        centroid_r = round(np.average([ x[0] for x in centroid_map[key]]))
        centroid_c = round(np.average([ x[1] for x in centroid_map[key]]))
        cv2.line(ret_img, (centroid_c-10,centroid_r), (centroid_c+10,centroid_r), (0, 0, 255), 5)
        cv2.line(ret_img, (centroid_c,centroid_r-10), (centroid_c,centroid_r+10), (0, 0, 255), 5)

cv2.imwrite('./output/3.jpg',ret_img)
```

IV. Results

a. a binary image (threshold at 128)

b. a histogram



資工所 r08922123
Author: 王韻豪

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