

電腦視覺 HW2 report

D10922012 資工所 王傳啟

Write a program to generate:

(a) a binary image (threshold at 128):

解釋: 迴圈掃整張圖片矩陣, 將值超過 128(含)以上的設定為 255 其他歸 0。

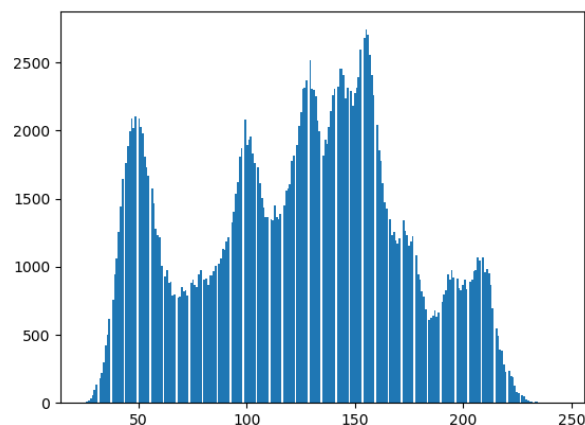
```
for i in range(row):
    for j in range(col):
        np_img_a[i][j] = 255 if(np_img_a[i][j] >= 128) else 0
```



(b) a histogram:

解釋: 此處用 dictionary 將值統計在一起, 而套件作圖時只需要用 list 即可。

```
hist_dict = dict()
hist_list = list()
np_img_b = copy.deepcopy(np_img)
for i in range(row):
    for j in range(col):
        if(np_img_b[i][j] not in hist_dict):
            hist_dict[np_img_b[i][j]] = 1
        else:
            hist_dict[np_img_b[i][j]] = hist_dict[np_img_b[i][j]] + 1
    hist_list.append(np_img_b[i][j])
```



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

解釋: 我是採用老師上課所教的**"The Classical Algorithm"**來完成此題, 且是使用**"4 連通"**的方式來判斷

分成以下步驟方便後面解釋:

1. 二值化圖片
2. 根據"左"與"上"的鄰居關係, 建立 記錄矩陣 與 相鄰配對
3. 參照 2022 講義, 將 相鄰配對 根據相鄰關係合併成 等價類 (equivalence classes)
4. 透過等價類, 重新將 記錄矩陣 的值更改成 等價類中最小者
5. 根據題目, 找出超過 500 個 pixels 以上的 等價類, 並加以作圖(矩形與重心)

1. 二值化圖片, 與 (a) 程式完全相同, 故略。

2. 根據"左"與"上"的鄰居關係, 建立 記錄矩陣 與 相鄰配對:

```
record_matrix = np.zeros((row, col), dtype = np.int)
number = 1
pair_set = set()
# step2: create a recordMatrix and the pair of the top and left
# the pair is add into the set
for i in range(row):
    for j in range(col):
        if np_img_c[i][j]: # if the pixel is white
            up = record_matrix[i-1][j] if i-1 >= 0 else 0
            left = record_matrix[i][j-1] if j-1 >= 0 else 0
            if not up and not left:
                record_matrix[i][j] = number
                number += 1
            else:
                if not up:
                    record_matrix[i][j] = left
                elif not left:
                    record_matrix[i][j] = up
                else:
                    pair_set.add((up, left))
                    record_matrix[i][j] = min(up, left)
```

3. 參照 2022 講義, 將 相鄰配對 根據相鄰關係合併成 等價類 (equivalence classes):

```
# step3: merge all pairs into a unique set, and add that set into a list
```

```
merged_list = list()
for i, ele in enumerate(pairs_list):
    newSet = {ele[0], ele[1]}
    if not merged_list:
        merged_list.append(newSet)
    else:
        frontSet = set()
        secondSet = set()
        for mSet in merged_list:
            if ele[0] in mSet:
                frontSet = mSet
            if ele[1] in mSet:
                secondSet = mSet
        if frontSet and secondSet:
            if frontSet == secondSet:
                merged_list.remove(frontSet)
            else:
                merged_list.remove(frontSet)
                merged_list.remove(secondSet)
        else:
            if frontSet:
                merged_list.remove(frontSet)
            elif secondSet:
                merged_list.remove(secondSet)
        newSet = newSet | frontSet | secondSet
        merged_list.append(newSet)
```

4. 透過等價類, 重新將 記錄矩陣 的值更改成 等價類中最小者:

```
# step4: merge the recordMatrix
# step4.1: use merged list to build a hashtable
merged_dict = dict()
for mSet in merged_list:
    mSet_list = list(mSet)
    for ele in mSet_list:
        merged_dict[ele] = min(mSet)
# step4.2: use the hashtable to merge the recordMatrix
for i in range(row):
```

```

        for j in range(col):
            # if recordMatrix has value, change to the smaller number
            if record_matrix[i][j] and record_matrix[i][j] in
merged_dict.keys():
                record_matrix[i][j] =
merged_dict[int(record_matrix[i][j])]

```

5. 根據題目, 找出超過 500 個 pixels 以上的 等價類, 並加以作圖(矩形與重心):

```

# step5: plot the picture
# count the number of the number of recordMatrix
count_dict = dict()
for i in range(row):
    for j in range(col):
        if record_matrix[i][j]:
            if record_matrix[i][j] not in count_dict.keys():
                count_dict[record_matrix[i][j]] = 1
            else:
                count_dict[record_matrix[i][j]] += 1
# filter the numbers which are over the requirement of the HW2
height_dict = dict()
width_dict = dict()
over500_list = list()
for key, value in count_dict.items():
    if value > 500: # the requirement of the HW2
        over500_list.append(key)
        height_dict[key] = list()
        width_dict[key] = list()
# put the elements into the list of the corresponding hashtable
for _, ele in enumerate(over500_list):
    for i in range(row):
        for j in range(col):
            if(record_matrix[i][j] == ele):
                height_dict[ele].append(i)
                width_dict[ele].append(j)
# create figure and axes
_, ax = plt.subplots()
img_c = Image.fromarray(np_img_c)
ax.imshow(img_c, cmap=plt.cm.gray, vmin=0, vmax=255)

```

```

# plot the rectangles and the centroids
for _, ele in enumerate(over500_list):
    max_h = max(height_dict[ele])
    min_h = min(height_dict[ele])
    max_w = max(width_dict[ele])
    min_w = min(width_dict[ele])
    mean_h = int(sum(height_dict[ele])/len(height_dict[ele]))
    mean_w = int(sum(width_dict[ele])/len(height_dict[ele]))
    # create a Rectangle patch
    rect = patches.Rectangle((min_w, min_h), max_w-min_w, max_h-
min_h,
                            linewidth=2, edgecolor='b', facecolor='none')
    # add the rectangle to the axes
    ax.add_patch(rect)
    # add the the centroid (+) to the axes
    plt.plot(mean_w, mean_h, marker='+', mew=4, ms=8, color='r')
plt.savefig('result/connected-components.png')

```

