R09922190 王祥任

CV Homework 2

Abstract

本次作業中，我們要實作(1)以128為界，進行圖片的二值化操作，(2)統計未二值化前圖片的灰度做成直方圖，(3)尋找二值化圖片白色區塊的邊界和其質心，並以500作為面積篩選的標準。以上操作除輸入、輸出以外，不得直接套用cv2等現成套件。

Implementation

- Programming Language: Python3
- Python Package: opencv-python, matplotlib, numpy, copy
- Execution: python3 hw2-main.py

在執行時，請務必將 lena.bmp 放置在和 hw2-main.py 相同的檔案目錄下，並確認檔名相同。opencv-python用於讀取和寫出圖片，matplotlib僅用於runtime時即時顯示和儲存圖片，須配合jupyter使用，在此不贅述，而最後的copy和numpy則用於純計算上。

Binarize at 128

```python
def binarize(lena, thr=128):
    after = copy.deepcopy(lena)
    for r in range(after.shape[0]):
        for c in range(after.shape[1]):
            if (after[r][c][0] < thr) : after[r][c] = [0,0,0]
            else: after[r][c] = [255,255,255]
    return after
```

遍歷原圖之像素，若像素之灰度為0~127，則歸類於白色[0,0,0]，而若為128~255則歸類於黑色[255,255,255]。imread預設以RGB讀入原圖，但原圖為灰階圖片，故三值會相同，取其中一通道來進行比較即可。最終輸出成果請參照報告最後的圖組。

Histogram of lena.bmp

```python
def get_historgram(lena):
    hist = np.zeros(256)
    for r in range(lena.shape[0]):
        for c in range(lena.shape[1]):
            hist[lena[r][c][0]] += 1
    return hist
hist = get_historgram(lena)
plt.clf()
plt.bar([i for i in range(256)], hist, width=1)
plt.savefig("histogram.png")
```

建立一個長度為256的零陣列，遍歷所有像素，並根據其灰度做為陣列的index，統計有多少像素屬於同一灰度，最終以matplotlib的bar來匯出histogram，最終輸出成果請參照報告最後的圖組。

Connected Components

```python
def FindSet(bina):
    height, width, _ = bina.shape

    ufs = UnionFindSet2D(height, width)
    four_dir = [
        np.array([-1, 0]),  # up
        np.array([1, 0]),  # down
        np.array([0, -1]),  # left
        np.array([0, 1]),  # right
    ]

    def is_outbound(coord, height, width):
        return (coord[0] < 0 or coord[1] < 0 or
                coord[0] >= height or coord[1] >= width)

    for r in range(height):
        for c in range(width):
            if (bina[r][c][0] != 255): # Not White
                continue

            cur = np.array([r, c])
            for d in four_dir:
                tmp = copy.deepcopy(cur) + d

                if (is_outbound(tmp, height, width)):
                    continue
                elif (bina[tmp[0]][tmp[1]][0] == 255):
                    ufs.union(cur, tmp)

    return ufs

def AggregateSet(bina, ufs):
    set_table = {}
    for r in range(bina.shape[0]):
        for c in range(bina.shape[1]):
            if (bina[r][c][0] == 255):
                tmp = r * bina.shape[1] + c
                root = ufs.find_by_idx(tmp)
                if (root in set_table):
                    set_table[root].append(np.array([r, c]))
                else:
                    set_table[root] = [np.array([r, c])]
    return set_table
```

```python
def FiltArea(set_table, limit=500):
    to_pop = []
    for key, val in set_table.items():
        if (len(val) < limit):
            to_pop.append(key)

    for key in to_pop:
        set_table.pop(key)

    return set_table

def DrawBoundaryAndCentroid(bina, filt_set_table):
    after = copy.deepcopy(bina)

    for key, pixel_set in filt_set_table.items():
        area = len(pixel_set)
        cen_r = 0
        cen_c = 0

        left, up= (lena.shape[1], lena.shape[0])
        right, down = (-1, -1)

        for pixel in pixel_set:
            if up > pixel[0]: up = pixel[0]
            if down < pixel[0]: down = pixel[0]
            if right < pixel[1]: right = pixel[1]
            if left > pixel[1]: left = pixel[1]

            cen_r += pixel[0]
            cen_c += pixel[1]

        cen_r = cen_r // area
        cen_c = cen_c // area

        # coordinate of cv2 is (x, y) = (column, row)
        cv2.rectangle(after, (left, up), (right, down), (0, 0, 255), 3)
        cv2.drawMarker(after, (cen_c, cen_r), (255, 0, 0), 0, thickness=2)
        cv2.circle(after, (cen_c, cen_r), 3, (255, 0, 0), 3)

    return after
```

首先在FindSet中，使用 Union Find Set (具體的實作方式請參考hw2-main.py)，四連通的方式，尋找以128為界，二值化圖片的白色(255,255,255)像素的連通群。遍歷所有白色像素，若其四個方向的像素也為白色，則將他們並聯（Union）起來。而此時，只要為相同群的像素，都會有相同的root，而AggregateSet便是再遍歷一次圖片，並找出所有白色像素的root，並依他們所屬的root歸類成一張表格。然後在FiltArea中，將同一root下，不滿500像素的群體從表格中刪除。我們將面積大小足夠的每一群體，遍歷其中所有的像素，找出邊界，同時也會加總像素位置row和column的值並最終除以群的面積大小，獲得質心的位置。最終將邊界和質心繪製在圖片上即完成。最終輸出成果請參照報告最後的圖組。

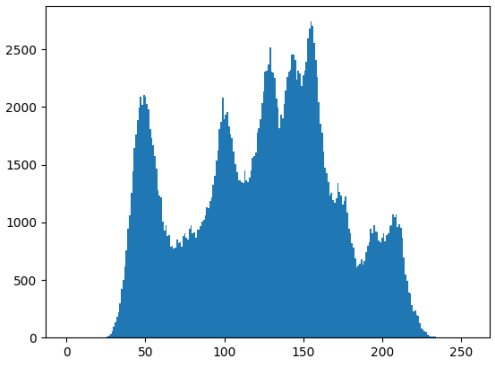
Results

下一頁的圖組中，左上角為原圖，右上角為以128為界進行二值化的結果，左下角為統計灰度的直方圖(histogram)，而右下角為以白色像素來找尋Connected Components和其質心的結果，藍色線為邊界，紅色十字為每一群的質心。因為排版的因素，所以圖片的大小有略為調整。

R09922190 王祥任

Original

Binarize at 128



Histogram of lena.bmp

Connected Components