NCLAB HW Day1

劉宇舜

Connected component labeling

- 目標是希望找出圖片 中的連通塊
- 需要克服的問題:
  - 遍歷所有檔案
  - 找出連通塊
  - 找出y座標最小的連通塊，可是考慮到雜訊，所以我們需要把面積納入考量
  - 生成資料夾

Disjointed set (DSU)

- 使用於找連通塊的演算法，適用於網路圖，考慮到二維的圖片也可以看成網路圖，所以適用這個演算法。
- 變數名稱意義：
  - 圖片長寬: r, c
  - 節點所屬連通塊 id: rt #意思是 root
  - 原圖片 : arr #(480, 640) 之二為陣列
  - 同連通塊中有多個 id ，每個id 所屬連通塊中的代表id(通常為最小id): boss
  - 建立邊的List: relation

```
def dsu(arr):
    [r, c] = arr.shape
    rt = np.zeros((r, c))
    rt = rt.astype('uint8')
    set_num = 1
    boss = [0]
    for i in range(r):
        for j in range(c):
            if arr[i][j] == 0:
                continue
            relation = []
            if i > 0 and rt[i-1][j] != 0:
                relation.append(dsu_find(boss, rt[i-1][j]))
            if j > 0 and rt[i][j-1] != 0:
                relation.append(dsu_find(boss, rt[i][j-1]))
            if len(relation) >= 2:
                dsu_union(boss, relation[0], relation[1])
            if len(relation) == 0:
                boss.append(set_num)
                relation.append(set_num)
                set_num += 1
            rt[i][j] = relation[0]
```

Disjointed set (DSU)

- Dsu 的必要兩個function
- Dus_find 找id所屬的連通塊中的代表id
- Dus_union 當需要連接兩個連通塊需要處理 id表

```
def dsu_find(boss, req):
    res = boss[req]
    if res == req:
        return res
    else:
        boss[req] = dsu_find(boss, res)
        return boss[req]

def dsu_union(boss, reqA, reqB):
        boss[reqB] = reqA
```

OpenCV 注意事項

- 由於我們使用 0~255來呈現灰階圖片 ，所以每個單位的型態需要是 unit8

```
rt_img = np.zeros((r, c))
rt_img = rt_img.astype('uint8')
```

檔案處理 注意事項

- 需要先建立資料夾，可是考慮到資料夾已存在，會建立失敗，所以需要使用 try&except

```
for root, dirs, files in walk(mypath):
    try:
        os.makedirs(root+'_mask')
    except:
        pass
    for it in files:
        if it.find(".png") == -1:
            continue
        else:
            fullpath = join(root, it)
            print(fullpath)
            img = cv2.imread(fullpath, cv2.IMREAD_GRAYSCALE)
            res = dsu(img)
            fullpath2 = join(root+'_mask', it)
            cv2.imwrite(fullpath2, res)
```