

CV HW6

B07902072 資工四 陳光裕

Usage

```
python3 main.py
```

corner

```
corner = [[[1,0],[1,-1],[0,-1]],[[0,-1],[-1,-1],[-1,0]],[[-1,0],[-1,1],[0,1]],[[0,1],[1,1],[1,0]]]
```

Corner Neighborhood
(for corresponding x_i)

	x_2	x_6
		x_1

x_7	x_2	
x_3		

x_3		
x_8	x_4	

		x_1
	x_4	x_5

流程敘述

- 首先先做Binary，然後對二值圖做Downsampling(以8 * 8的block中最左上角的值作為整個block的值)，得到64 * 64的新圖。

```
def down_sampling(img):  
    width, height = 512, 512  
    new_img = np.zeros((64,64))  
    for i in range(64):  
        for j in range(64):  
            new_img[i][j] = img[i*8][j*8] # left top point = new point  
    return new_img # new img is a 64*64 array
```

- 接著計算Yokoi connectivity number，我用3個迴圈處理，前兩個迴圈處理64 * 64的所有點，第三個迴圈計算每個點的corner neighbor，遇到邊界則設為0。

```

for i in range(width):
    row = []
    for j in range(height):
        if img[i][j] == 0:
            # print space
            row.append(' ')
            continue
        else:
            # check every corner's h_equation then check qrs's number
            cnt = []
            for cor in corner:
                # four corner test
                b = img[i][j]
                c = yokoi_conn(img, i, j, width, height, cor[0])
                d = yokoi_conn(img, i, j, width, height, cor[1])
                e = yokoi_conn(img, i, j, width, height, cor[2])
                cnt.append(h_equation(b, c, d, e))

```

```

def valid(x, y, w, h):
    return x >= 0 and x < w and y >= 0 and y < h

def yokoi_conn(img, x, y, w, h, cor):
    new_x = x+cor[0]
    new_y = y+cor[1]
    if valid(new_x, new_y, w, h):
        return img[new_x][new_y]
    return 0

```

- 對每個corner+自己本身去做h equation，如講義上的做法，然後將得到的結果append進一個陣列。

- for 4-connectivity

$$h(b, c, d, e) = \begin{cases} q & \text{if } b = c \text{ and } (d \neq b \vee e \neq b) \\ r & \text{if } b = c \text{ and } (d = b \wedge e = b) \\ s & \text{if } b \neq c \end{cases}$$

```

def h_equation(b, c, d, e):
    if b == c and (d != e or e != b): return 'q'
    if b == c and (d == b and e == b): return 'r'
    if b != c: return 's'

```

- 最後計算r跟q的數量，得到yokoi connectivity number，最後append進答案的陣列。

$$f(a_1, a_2, a_3, a_4) = \begin{cases} 5 & \text{if } a_1 = a_2 = a_3 = a_4 = r \\ n & \text{where } n = \text{number of } \{a_k | a_k = q\}, \text{ otherwise} \end{cases}$$

```
# r = 4 : 5 , else print q's number
if cnt.count('r') == 4:
    row.append('5')
else:
    if cnt.count('q') == 0:
        row.append(' ')
    else:
        row.append(str(cnt.count('q')))
```

- 最後把答案陣列儲存成一個txt檔。

```
def save(result):
    file = open('output.txt', 'w')
    with open('output.txt', 'w') as file:
        for i in range(result.shape[0]):
            for j in range(result.shape[1]):
                file.write(result[i][j])
            file.write('\n')
```

Result

```
b07902072@linux8 [~/CV/HW6] cat output.txt
11111111 12111111111122322221 111111111111
15555551 115555555511 2 11 11 115555555511
15555551 1 2115555112 21112221 155555555551 21
15555551 1 2 155112 22221511 155555555551 1
15555551 22 2112 22 121 155555555551
15555551 1 2 21 2 1 1 1555555555551
15555551 12 1 121111 1321 15555555555511
15111551 1322 1155551111 15555555555551
111 1551 1 121555555511 155555555555511
11 1551 21155555511 1551115555511
21 1551 2 15555555111 1551 11555511
1 1551 2 155555555511 1551 115551 1
1551 1121155555555551 1551 15511 12
1551 1555555555555511 1551 1111 111
1551 1 222115555555555511 1151 11 1151
1551 2 22 1 1555555555555511 151 11111 1551
1551 2 1 11555555555555551 151 115551 11551
1551 2 11555555555555555111511155511 115551
1551 12 1155555555555555555555555551 155551
1551 11 221555555555555555555555555112 1155551
1551 111 22 15555555555555555555555551 1 1555551
1551 1511 1 12511211111211155555555111 11555551
1551 15521 1 121 1 11 1 15555555111 15555551
1551 1151 132 2 1155555111 115555551
1551 151 322 115555111 121 155555551
1551 1221 2 1555551 131 1155555551
1551 2 1 115555511 1 1155555551
1551 2 1155555551 1 155555551
1551 2 1155555551 21155555551
1551 1 11555555551 1555555551
1551 1 11511115555521 1 11555555551
1551 1 1 11111 1155511 2 155555555551
1551 131 111 15111 2 155555555551
1551 121 1121 1 111 1 2 115555555551
1551 11 111 1 221 11 1 2 155555555551
1551 12 1 21 121 11 1111 2 155555555551
1551 1 12 22 151111111551 2 11555555555551
1551 1 2 1555551115511 1 15555555555551
1551 2 22 12555551 15551 1 15555555555551
1551 1 1 1555511 11511 2 11555555555551
1551 21 155551 1 151 2 15555555555551
1551 2 15555112 151 2 15555555555551
1551 1 1 1 1155555511111 2 15555555555551
1551 2 22 111511111212 2115555555555551
1551 1 12 151 2 1 15555555111555551
1551 1111 121 155555551 1555551
1551 11111111 155555551 1555551
1551 115551 155555551 1555511
1551 15551 211111111 155511
11521 1 12 122155511 2 11 115511
1 151 1 1 155555111 2111 15511
22 1511 1 15555555111 155111 1511
22 1511 1 15555555551 155551 1151
2 151 1 11155555555511 155511 1511
2 1521 1 155555555555511 15551 12151
2 151 121 15555555555551 155511 1551
2 1511 155555555555551 115551 1511
21 1511 11 15555555555551 111111151
11 151 1155555555555551 111511
11 151 1555555555555551 151
11 151 1155555555555551 211
11 151 11555555555555511 1
11 151 1555555555555551
11 111 1211111111111111111
```