

CV HW6

R11922196 林佑鑫

(a) Yokoi connectivity number

Brief description, algorithm:

Step 1: 用 HW1 的 shrink 跟 HW2 的 binarize function 做 down-sampling。

Step 2: 依照講義 6.2.5.1 的公式實作 h_function 與 f_function。

Step 3: 輸出成 txt 文檔。

Parameters: None

Principal code fragment:

Step 1:

```
4  def binarize(img):
5      height, width = img.shape
6      binarize_img = np.zeros_like(img)
7
8      for h in range(height):
9          for w in range(width):
10             if((img[h, w] >= 128)):
11                 binarize_img[h, w] = 255
12             else:
13                 binarize_img[h, w] = 0
14      return binarize_img
15
16 def shrink(img, ratio):
17     height, width = img.shape
18     new_height = height // ratio
19     new_width = width // ratio
20     shrink_img = np.zeros((new_height, new_width))
21
22     for h in range(new_height):
23         for w in range(new_width):
24             shrink_img[h, w] = img[h * ratio, w * ratio]
25
26     return shrink_img
```

Step 2:

```

28 def yokoi_connectivity(img):
29     height, width = img.shape
30
31     pad_img = np.zeros((height+2, width+2)) - 1
32     pad_img[1:height+1, 1:width+1] = img
33
34     output_img = np.zeros_like(img) - 1
35
36     for h in range(1, height+1):
37         for w in range(1, width+1):
38             if pad_img[h, w] == 255:
39                 a1 = h_function(pad_img[h, w], pad_img[h, w+1], pad_img[h-1, w+1], pad_img[h-1, w])
40                 a2 = h_function(pad_img[h, w], pad_img[h-1, w], pad_img[h-1, w-1], pad_img[h, w-1])
41                 a3 = h_function(pad_img[h, w], pad_img[h, w-1], pad_img[h+1, w-1], pad_img[h+1, w])
42                 a4 = h_function(pad_img[h, w], pad_img[h+1, w], pad_img[h+1, w+1], pad_img[h, w+1])
43
44                 output_img[h-1, w-1] = f_function(a1, a2, a3, a4)
45
46     return output_img
47
48 def h_function(b, c, d, e):
49     if b == c and (b != d or b != e):
50         return 'q'
51     elif b == c == d == e:
52         return 'r'
53     else:
54         return 's'
55
56 def f_function(a1, a2, a3, a4):
57     if a1 == a2 == a3 == a4 == 'r':
58         return 5
59     else:
60         return [a1, a2, a3, a4].count('q')

```

Step 3:

```

73     height, width = yokoi_map.shape
74     with open('yokoi.txt', 'w') as file:
75         for h in range(height):
76             for w in range(width):
77                 if yokoi_map[h, w] != -1:
78                     file.write(f'{int(yokoi_map[h, w])}')
79                 else:
80                     file.write(f' ')
81             file.write('\n')

```

Resulting image:

```

11111111 1211111111112232221 11111111111 0 0
15555551 11555555511 2 11 11 15555555511 0
15555551 1 21555112 21122221 1555555551 21
15555551 1 2 155112 22221511 15555555511 1
15555551 22 2112 22 121 0 0 15555555511 0
15555551 1 2 21 2 1 155555555551 0
15555551 12 1 12111 1321 15555555511
15111551 1322 1155551111 155555555551
111 1551 1 12155555511 1555555555511
11 1551 215555551 15511555551
21 1551 2 155555511 1551 1155511
1 1551 2 15555555511 1551 115551
1551 11211555555551 1551 15511 12
1551 1 222115555555511 1551 1111 111
1551 2 22 1 155555555555511 151 11111 1551
1551 2 1 1155555555555551 151 115551 11551
1551 2 1155555555555551151115551 115551
1551 12 1155555555555555555551 15551
1551 11 0 22155555555555555555555112 115551
1551 111 22 1555555555555555555551 1 155551
1551 1511 1 1251121111121115555555111 1155551
1551 15521 1 121 1 11 1 155555511 0 155551
1551 1151 132 2 115555511 0 11555551
1551 151 0 322 11555511 121 15555551
1551 1221 2 155551 131 11555551
1551 2 0 1 1155551 1 11555551
1551 2 0 0 115555551 0 1 15555551
1551 2 1155555551 2115555551
1551 1 0 11555555551 1555555551
1551 1 115111155551 1 11555551
1551 1 1 11111 1155511 2 1555555551
1551 131 111 15111 2 1555555551
1551 121 0 1121 1 111 1 2 115555555551
1551 11 111 1 221 11 2 155555555551
1551 12 0 1 21 121 11 1111 2 155555555551
1551 1 0 12 22 15111111551 2 115555555551
1551 1 0 0 2 15555115511 1 155555555551
1551 2 0 0 22 1255551 15551 1 155555555551
1551 1 0 0 1 1555511 11511 2 115555555551
1551 0 0 21 155551 1 151 2 15555555555551
1551 1 0 1555512 151 2 1555555555551
1551 1 1 1 115555511111 2 15555555555551
1551 2 22 111511111212 21155555555551
1551 0 1 12 151 2 1 15555551155551
1551 0 0 0 1111 121 1555551 155551
1551 11111111 15555551 155551
1551 0 115551 15555551 1555511
1551 15521 1 12 12215511 2 11 15511
1 151 0 1 1 15555511 2111 15511
22 1511 1 1555555511 155111 1511
22 1511 1 1555555551 155551 1551
2 151 0 1 1115555555511 155511 1511
2 1521 0 1 15555555555511 15551 12151
2 151 121 15555555555551 155511 1551
2 1511 0 11 15555555555551 115551 1511
21 1511 11111151 11111151
11 151 0 11555555555551 111511
11 151 1555555555555551 151
11 151 0 1155555555555551 211
11 151 1155555555555551 1
11 151 0 1555555555555551
11 111 0 1211111111111111

```