

Homework 6

R13525009 羅筠笙

a. Yokoi Connectivity Number

Description: Processes a grayscale Lena image by binarizing it, then downsampling it to a 64x64 grid using the top-left pixel from each 8x8 block. It calculates the Yokoi connectivity number for each pixel in the downsampled binary image to determine its connectivity pattern. Finally, it displays the result as text on a 64x64 grid, where each number represents the connectivity value for each pixel location.



```
1. from PIL import Image as im
2. import numpy as np
3. import matplotlib.pyplot as plt
4.
5. img = im.open("./lena.bmp")
6.
7. img_arr_binary = (img_arr >= 128).astype(np.uint8) * 255
8.
9. downsample = np.zeros((64, 64), dtype=np.uint8)
10. for i in range(64):
11.     for j in range(64):
```

```

12.         downsample[i, j] = img_arr_binary[i * 8, j * 8]
13.
14. # Yokoi Connectivity
15. def yokoi_connectivity(pixel, img, x, y):
16.     def h(b, c, d, e):
17.         if b == c and (b != d or b != e):
18.             return 1
19.         elif b == c == d == e:
20.             return 2
21.         else:
22.             return 0
23.
24.     if pixel == 0:
25.         return 0
26.
27.     neighbors = {
28.         "r": img[x, y+1] if y + 1 < 64 else 0,
29.         "q": img[x-1, y] if x - 1 >= 0 else 0,
30.         "s": img[x, y-1] if y - 1 >= 0 else 0,
31.         "t": img[x+1, y] if x + 1 < 64 else 0
32.     }
33.
34.     # Count number
35.     count = sum([
36.         h(pixel, neighbors["r"], neighbors["t"], neighbors['q']),
37.         h(pixel, neighbors["q"], neighbors["s"], neighbors['r']),
38.         h(pixel, neighbors["s"], neighbors["t"], neighbors['q']),
39.         h(pixel, neighbors["t"], neighbors["r"], neighbors['s'])
40.     ])
41.
42.     return 5 if count == 4 else count
43.
44. result = np.full((64, 64), " ", dtype=str)
45. for i in range(64):
46.     for j in range(64):
47.         result[i, j] = str(yokoi_connectivity(downsample[i, j], downsamp
le, i, j))

```