

I. Problem Statement

- Write a program which counts the Yokoi connectivity number on a down-sampled image(lena.bmp).

II. Programming Tools

- Programming language: Python 3.8.5
- Library: Numpy 1.19.1, OpenCV 4.0.1

III. Problem-Solving Process

用 `cv2.imread('lena.bmp',0)` 讀取灰階影像，使用 `threshold=127` binarize 灰階影像得到 `img`，`img` 大小為(512,512)，之後再對 `img` 做 down sampling，得到(64,64)的 `img`，最後使用四連通作為鄰居條件，依序對 `img` 中的每個 pixel 做檢測，並計算出每個 pixel 對應的 Yokoi connectivity number，再儲存成.txt 檔，完成本次作業。

```
import numpy as np
import cv2

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

def f(a1,a2,a3,a4):
    return 5 if a1 == a2 == a3 == a4 == 'r' else [a1,a2,a3,a4].count('q')

def Yokoi_connectivity_number(img):
    img = np.where(img>127, 255, 0) # binarilize
    img = img[::8,::8] # down sampling
    img_padding = np.pad(img, ((2,2),(2,2)), 'constant',constant_values = (0,0))
    ret = np.full_like(img, ' ', dtype='<U1')
    for r in range(img.shape[0]):
        for c in range(img.shape[1]):
            if img[r,c] != 255:
                continue
            a1 = h(img_padding[r+2,c+2], img_padding[r+2,c+3], img_padding[r+1,c+3], img_padding[r+1,c+2])
            a2 = h(img_padding[r+2,c+2], img_padding[r+1,c+2], img_padding[r+1,c+1], img_padding[r+2,c+1])
            a3 = h(img_padding[r+2,c+2], img_padding[r+2,c+1], img_padding[r+3,c+1], img_padding[r+3,c+2])
            a4 = h(img_padding[r+2,c+2], img_padding[r+3,c+2], img_padding[r+3,c+3], img_padding[r+2,c+3])
            val = f(a1,a2,a3,a4)
            ret[r,c] = str(val) if val else ' '
    return ret

if __name__ == "__main__":
    img = cv2.imread('lena.bmp',0)
    np.savetxt('./output/Yokoi_connectivity_number.txt', Yokoi_connectivity_number(img), fmt='%s')
```

