

Computer Vision Homework 6

R08922079 資工所一 洪浩翔

Part 0

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
import time

def readImg(filename='lena.bmp'):
    #read img
    image = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
    print('shape:', image.shape)
    #binarize
    index = np.where(image >= 128)
    binary = np.zeros(image.shape)
    binary[index] = 255
    cv2.imwrite('binary.jpg', binary)
    return image, binary

def downSample(image):
    #down sample to 64*64
    w,h = image.shape
    result = np.zeros((int(w/8), int(h/8)))
    for i in range(0,w,8):
        for j in range(0,h,8):
            result[int(i/8),int(j/8)] = image[i,j]
    cv2.imwrite('downSample.jpg', result)
    return result
```

Pre-processing and down-sampling code fragment

Part 1

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

11111111	12111111111122322221	111111111111	0 0
15555551	115555555511 2 11 11	115555555511	0
15555551	1 2115555112 21112221	155555555551	21
15555551	1 2 155112 22221511	155555555511	1
15555551	22 2112 22 121 0 0	1555555555511	0
15555551	1 2 21 2 1 1	1555555555551	0
15555551	12 1 121111 1321	15555555555511	
15111551	1322 1155551111	15555555555551	
111 1551	1 121555555511	15555555555511	
11 1551	21155555511	1551115555511	
21 1551	2 15555555111	1551 11555511	
1 1551	2 155555555511	1551 115551	1
1551	112115555555551	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 115555555555555555555555555555551		155551
1551	11 0 22155555555555555555555555555555112		1155551
1551	111 22 155555555555555555555555555555551 1		1555551
1551	1511 1 12511211111211155555555555111		11555551
1551	15521 1 121 1 11 1 15555555111 0		15555551
1551	1151 132 2 11555555111 0		115555551
1551	151 0 322 115555111 121		155555551
1551	1221 2 1555551 131		1155555551
1551	2 0 1 115555511 1		1155555551
1551	2 0 0 1155555551 0		1 155555551
1551	2 11555555551		21155555551
1551	1 0 115555555551		15555555551
1551	1 11511115555521 1		115555555551
1551	1 1 11111 1155511 2		155555555551
1551	131 111 15111 2		155555555551
1551	121 0 1121 1 111 1 2		1155555555551
1551	11 111 1 221 11 1 2		1555555555551
1551	12 0 1 21 121 11 1111 2		1555555555551
1551	1 12 22 151111111551 2		11555555555551
1551	1 2 1555551115511 1		15555555555551
1551	2 0 0 22 12555551 15551 1		155555555555551
1551	1 1555511 11511 2		1155555555555551
1551	0 0 21 155551 1 151 2		1555555555555551
1551	2 15555112 151 2		1555555555555551
1551	1 1 1 1155555511111 2		1555555555555551
1551	2 22 111511111212 211555555555555551		211555555555555551
1551	0 1 12 151 2 1 15555555111555551		155555555115555551
1551	0 0 0 1111 121 155555551 1555551		1555555551 1555551
1551	0 11111111 155555551 1555551		1555555551 1555551
1551	0 15551 155555551 1555511		1555555551 1555511
1551	15551 211111111 155511		211111111 155511
11521	1 12 122155511 2 11 115511		2 11 115511
1 151 0	1 1 155555111 2111 15511		155555111 15511
22 1511	1 15555555111 155111 1511		1555555551 15511
22 1511	1 15555555551 155551 1151		15555555551 15511
2 151	0 1 111555555555511 155511 1511		155555555555511 155511 1511
2 1521	0 1 1555555555555511 15551 12151		1555555555555551 155511 1551
2 151	121 1555555555555551 115551 1511		1555555555555551 115551 1511
2 1511	0 1555555555555551 115551 1511		1555555555555551 111111151
21 1511	11 1555555555555551 111511		1155555555555551 151
11 151	0 1155555555555551 151		1555555555555551 211
11 151	0 1155555555555551 1		1555555555555551 1
11 151	0 1555555555555551		1555555555555551
11 111	0 121111111111111111		

Yokoi connectivity number of down-sampled Lena

```

def hFunction(b, c, d, e):
    if b == c and (d != b or e != b):
        return "q"
    elif b == c and (d == b and e == b):
        return "r"
    else:
        return "s"

def counter(record):
    countQ = 0
    countR = 0
    for i in record:
        if i == "q":
            countQ += 1
        elif i == "r":
            countR += 1
    if countR == 4:
        return 5
    else:
        return countQ

```

$h(b,c,d,e)$ and counter for Yokoi connective number

```

def yokoi(image):
    image = np.pad(image, ((1,1),(1,1)), 'constant', constant_values=0)
    print('padding shape:', image.shape)
    w, h = image.shape
    with open('output.txt', "w") as f:
        for i in range(1, w-1):
            for j in range(1, h-1):
                if image[i,j] == 255:
                    record = []
                    record.append(hFunction(image[i,j], image[i+1,j], image[i+1,j-1], image[i,j-1]))
                    record.append(hFunction(image[i,j], image[i,j-1], image[i-1,j-1], image[i-1,j]))
                    record.append(hFunction(image[i,j], image[i-1,j], image[i-1,j+1], image[i,j+1]))
                    record.append(hFunction(image[i,j], image[i,j+1], image[i+1,j+1], image[i+1,j]))
                    num = counter(record)
                    f.write(str(num))
                else:
                    f.write(' ')
            f.write('\n')

```

Code fragment for Yokoi

```
if __name__ == "__main__":  
    image, binary = readImg()  
    down = downSample(binary)  
    yokoi(down)
```

Main()