電腦視覺 Homework 6 – Yokoi Connectivity Number

資工碩二 r08922a04 林承德

Language : python 3.7
Library :

```
import numpy as np
from PIL import Image
```

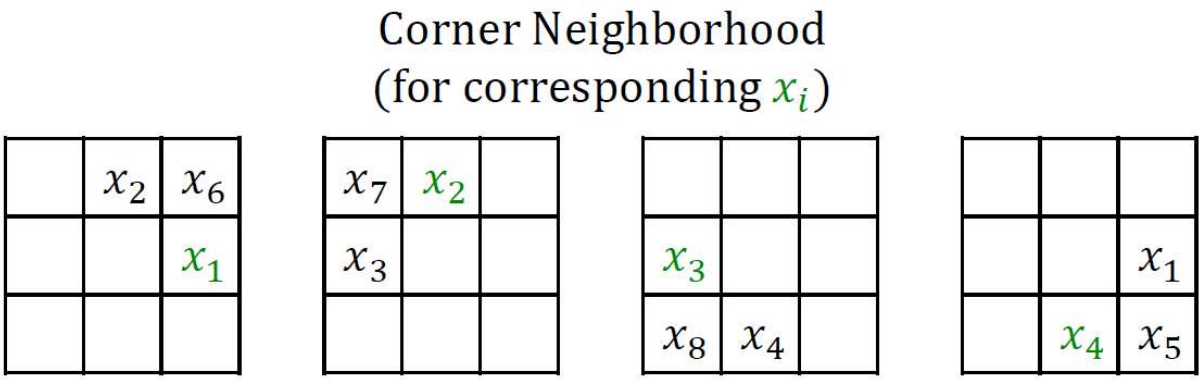
Execution way: python3 hw6.py
(please put lena.bmp at the same directory as hw6.py)

Problem: Calculate Yokoi Connectivity number

Description：

首先，先將原圖做downsampling(以最左上角的點代表整個8*8的block)後，對圖像進行前幾次作業已經做過很多次的binarization。這邊的順序有稍微調換一下因為我覺得先做downsampling可以節省一些計算量，binarization只要做downsample過的。

接著，就可以進行yokoi connectivity number 的計算了，我用兩個for迴圈，去看每個pixel，根據ppt上的 corner neighborhood 找出他的corner neighbor，若遇到邊界沒有值得時候設為0。



然後再利用ppt給的公式

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

待每個pixel 的b,c,d,e 計算出來後，就可以利用h equation，計算出q, r, s，透過結果便可以計算出該pixel的yoke connectivity number

Code

```python
from PIL import Image
import numpy as np

class Solution:
    def __init__(self):
        pass

    def binarization(self, image):

    def down_sampling(self, image, origin_size, target_size):
        pixels = image.load()
        new_image = Image.new(image.mode, (target_size, target_size))
        new_pixels = new_image.load()
        for i in range(target_size):
            for j in range(target_size):
                new_pixels[i, j] = pixels[i * int(origin_size/target_size), j * int(origin_size/target_size)]
        return new_image


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

    def valid(self, new_x, new_y, h, w):
        return new_x >= 0 and new_x < h and new_y >= 0 and new_y < w

    def check(self, image, cor, x, y):
        h, w = image.size
        pixels = image.load()
        new_x, new_y = x + cor[0], y + cor[1]
        if self.valid(new_x, new_y, h, w): return pixels[new_x, new_y]
        return 0
```

```python
def Yokoi_num(self, image, corners):
    pixels = image.load()
    res = []
    h, w = image.size
    for y in range(w):
        row = []
        for x in range(h):
            if pixels[x, y] == 0:
                row.append(' ')
                continue
            else:
                temp = []
                for cor in corners:
                    b = pixels[x, y]
                    c, d, e = 0, 0, 0
                    c = self.check(image, cor[0], x, y)
                    d = self.check(image, cor[1], x, y)
                    e = self.check(image, cor[2], x, y)
                    temp.append(self.h(b,c,d,e))
                if temp.count('r') == 4:
                    row.append('5')
                else:
                    if temp.count('q') == 0:
                        row.append(' ')
                    else:
                        row.append(str(temp.count('q')))
        res.append(row)
    return np.array(res)
```

```python
def save(self, save_name, res):
    file = open(save_name, "w")
    with open(save_name, "w") as file:
        for i in range(res.shape[0]):
            for j in range(res.shape[1]):
                file.write(res[i, j])
            file.write('\n')


def problem1(self, path, out_file, corners):
    image = Image.open(path)
    down_sampleing_image = self.down_sampling(image, 512, 64)
    # down_sampleing_image.save('down_sampling.bmp')
    bin_image = self.binarization(down_sampleing_image)
    # bin_image.save('bin_image.bmp')
    yokoi = self.Yokoi_num(bin_image, corners)
    self.save(out_file, yokoi)
```

Result: