電腦視覺 Homework 7 – Thinning

資工碩二 r08922a04 林承德

Language : python 3.7
Library :

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

Execution way: python3 hw7.py
(please put lena.bmp at the same directory as hw7.py)

Problem: Thinning

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過後64*64的。

接著分別根據公式，實作yokoi, pair relation, connected shrink

>> yokoi

和hw6相同，不多作說明

>> pair relation

利用兩層for loop 掃過每個pixel，根據yokoi的結果，與講義公式生成marked image

H function: (m="1", means "edge" in Yokoi)

- $h(a,m) = \begin{cases} 1, if\ a = m \\ 0, otherwise \end{cases}$

Output:

- $y = \begin{cases} q, if\ \sum_{n=1}^{4} h(x_n, m) < 1\ or\ x_0 \neq m \\ p, if\ \sum_{n=1}^{4} h(x_n, m) \geq 1\ and\ x_0 = m \end{cases}$

```
def pair_relationship(self, yokoi, image):
    marked = Image.new('1', image.size)
    marked_pixels = marked.load()
    h, w  = image.size
    for y in range(h):
        for x in range(w):
            if yokoi[y][x] == '1':
                booling = False
                for neighbor in self.neighbors:
                    new_x, new_y = x + neighbor[0], y + neighbor[1]
                    if self.valid(new_x, new_y, h, w):
                        if yokoi[new_y][new_x] == '1':
                            booling = True
                            break
                if booling:
                    marked_pixels[x,y] = 1
    return marked
```

>> connected shrink

利用兩層for loop 掃過marked image 的每個pixel，並檢查corner

H function: (yokoi corner => “q”)

- $h(b,c,d,e) = \begin{cases} 1, if\ b = c\ and\ (d \neq b\ or\ e \neq b) \\ 0, otherwise \end{cases}$

Output:

- $f(a_1,a_2,a_3,a_4,x) = \begin{cases} g, if\ exactly\ one\ of\ a_n = 1, n = 1{\sim}4 \\ x, otherwise \end{cases}$

```python
def connected_shrink(self, image, marked):
    pixels = image.load()
    marked_pixels = marked.load()
    res = image.copy()
    res_pixels = res.load()
    h, w = image.size
    for y in range(w):
        for x in range(h):
            if marked_pixels[x, y] == 1:
                temp = []
                for cor in self.corners:
                    b = res_pixels[x, y]
                    c, d, e = 0, 0, 0
                    c = self.check(res, cor[0], x, y)
                    d = self.check(res, cor[1], x, y)
                    e = self.check(res, cor[2], x, y)
                    temp.append(self.h_shrink(b, c, d, e))
                if temp.count(1) == 1:
                    res_pixels[x, y] = 0
    return res
```

（接下來用的方法是參考資料中所提供的，與上課內容稍有差異）

和上次的作業不同的是，上次只需要做一次的yokoi 就可以輸出結果了，但這次作業我們要重複的執行以下動作，直到結果收斂

重複動作1: run yokoi

重複動作2: run pair relation

重複動作3: run connected shrink

```python
def thinning(self, image):
    while True:
        yokoi = self.Yokoi(image)
        marked = self.pair_relationship(yokoi, image)
        thinning = self.connected_shrink(image, marked)
        if image != thinning:
            image = thinning
        else:
            break
    return thinning

def problem1(self, path, outfile):
    image = Image.open(path)
    down_sampled = self.down_sampling(image, 64)
    bin_image = self.binarization(down_sampled)
    thined = self.thinning(bin_image)
    print(thined)
    thined.save(outfile)
```

Result: