Computer Vision Homework 6

R06922152 袁晟峻

Yokoi Connectivity Number

1. Binarize the benchmark image Lena as in HW2 by 128

```
def binarize(image):
    width, height = image.size
    binaryImage = Image.new('1', (width, height), 'white')
    pixels = image.load()

    for j in range(height):
        for i in range(width):
            # threshold 128
            if pixels[i, j] < 128:
                binaryImage.putpixel((i, j), 0)
            else:
                binaryImage.putpixel((i, j), 1)
    return binaryImage
```

2. using 8x8 blocks as a unit, take the topmost-left pixel as the down-sampled data, Down-sample Lena from 512x512 to 64x64

```
def downSampling(image):
    output = Image.new('1', (64, 64))
    width, height = image.size
    pixels = image.load()

    for j in range(64):
        for i in range(64):
            output.putpixel((i, j), pixels[i*8, j*8])
    return output
```

取每 8x8 矩形的左上角存到新的圖，每次跳 8 格到下一個矩形。

3. Count the Yokoi connectivity number

```python
def countYokoiNumber(image):
    width, height = image.size
    output = Image.new('I', (width, height), 0)
    pixels = image.load()

    # initialize a bigger image
    bigImage = []
    for j in range(height+2):
        tmp = [0] * (width+2)
        bigImage.append(tmp)

    for i in range(1, height+1):
        for j in range(1, width+1):
            bigImage[i][j] = pixels[j-1, i-1]
```

這一部分我先建一個大一點的圖(66x66)，將原本

64x64 的 pixels 值存在裡面，外圍一圈 assign 成 0 以方

便接下來的計算。

```python
for i in range(1, height+1):
    for j in range(1, width+1):
        qCounter = 0
        rCounter = 0
        # a1
        if (bigImage[i][j] == 1):
            if bigImage[i][j] == bigImage[i+1][j] and bigImage[i][j] == bigImage[i][j-1] and bigImage[i][j] == bigImage[i+1][j-1]:
                rCounter += 1
            elif bigImage[i][j] == bigImage[i+1][j]:
                qCounter += 1
            # a2
            if bigImage[i][j] == bigImage[i][j-1] and bigImage[i][j] == bigImage[i-1][j-1] and bigImage[i][j] == bigImage[i-1][j]:
                rCounter += 1
            elif bigImage[i][j] == bigImage[i][j-1]:
                qCounter += 1
            # a3
            if bigImage[i][j] == bigImage[i-1][j] and bigImage[i][j] == bigImage[i][j+1] and bigImage[i][j] == bigImage[i-1][j+1]:
                rCounter += 1
            elif bigImage[i][j] == bigImage[i-1][j]:
                qCounter += 1
            # a4
            if bigImage[i][j] == bigImage[i][j+1] and bigImage[i][j] == bigImage[i+1][j] and bigImage[i][j] == bigImage[i+1][j+1]:
                rCounter += 1
            elif bigImage[i][j] == bigImage[i][j+1]:
                qCounter += 1

            if rCounter == 4:
                output.putpixel((j-1, i-1), 5)
            elif qCounter != 0:
                output.putpixel((j-1, i-1), qCounter)

return output
```

這邊真正計算 Yokoi number，對每一個 pixel 值為 1 的

pixel，其 connect number 由四個小方形決定，所以分成 a1, a2, a3, a4 4 個 if else，然後每一個去看是 r 或 q(或 s，但因為藉由 r 跟 q 就能判斷，故沒特別計算 s)，四個小方形結果若得到 4 個 r，則 connectivity number = 5，否則則看得到幾個 q，number 就是多少。

4. Result of this assignment is a 64x64 matrix.
   Please **align** the matrix within 1 single A4 page (using 4-connected).