Computer Vision HW6

R06922075 翁瑋

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

```python
def binarize(im , threshold) :
    pixels = im.load()
    im_thre = im.copy()

    for i in range(int(im.size[0])) :
        for j in range(int(im.size[1])) :
            im_thre.putpixel((i,j), (pixels[i,j] > threshold)*255 )

    return im_thre
```

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

```python
def down_sampling(im) :
    pixels = im.load()
    im_down = Image.new('L', (66,66) , 'black')

    for i in range(1 , im.size[0]+1 , 8) :
        for j in range(1 , im.size[1]+1 , 8) :
            im_down.putpixel((int(i/8) , int(j/8)) , pixels[i,j])
    return im_down
```

在這裡，為了處理 Yokoi 邊界上的問題，在最外圈加了一層 0

3. Count the Yokoi connectivity number 4. Result of this assignment is a 64x64 matrix.

```python
def yokoi(im) :
    pixels = im.load()

    yokoi_list = []
    for j in range(1,im.size[0]-1) :
        tmp = []
        for i in range(1,im.size[1]-1) :
            if pixels[i,j] == 255 :
                s = 0
                s += h(255 , pixels[i,j+1] , pixels[i-1,j+1] , pixels[i-1,j])
                s += h(255 , pixels[i-1,j] , pixels[i-1,j-1] , pixels[i,j-1])
                s += h(255 , pixels[i,j-1] , pixels[i+1,j-1] , pixels[i+1,j])
                s += h(255 , pixels[i+1,j] , pixels[i+1,j+1] , pixels[i,j+1])
                if s == 0.4 :
                    tmp.append(5)
                else :
                    tmp.append(int(s))
            else :
                tmp.append(' ')
        yokoi_list.append(tmp)

    return yokoi_list
```

對每個點的上下左右計算 Yokoi number，每一個 ax 都被丟進 h 做計算

```
46  def h(b,c,d,e) :
47      if b == c and b == d and b == e :
48          return 0.1
49      elif b != c :
50          return 0
51      else :
52          return 1
```

h(b,c,d,e) : 計算代表的是 q , r 或 s

Result :