

## Homework 6

資工所碩一 R05922068 彭宇劭

1. Write a program to generate Yokoi connectivity number

source code: hw6.py

執行方式：python hw6.py

版本：Python 2.7.10

Output：yokoi\_num.txt

簡述：

1. 在main function呼叫不同 function
2. 首先先將圖片二值化 (Threshold=128)
3. 將二值化後的圖片 downsampling  
原始圖512\*512 pixels，利用8\*8的小區域做 downsampling 取左上角的 pixel為代表，其後會產生64\*64的縮小圖

```
def down_sampling(img, k_size):  
    o_size = img.shape[0]  
    n_size = o_size/k_size  
    down_img = np.zeros((n_size, n_size), dtype=int)  
    for i in range(n_size):  
        for j in range(n_size):  
            down_img[i][j] = 1 if img[i*k_size][j*k_size]==255 else 0  
    return down_img
```

4. 計算每個pixel的 Yokoi Connectivity Number

呼叫 yokoi function 由上而下由左而右跑過整張64\*64 pixels，如果該點值為1則呼叫該點的四個角落計算h值，再根據h值的加總判斷該點的數值。

這邊將r設為10，q設為1，s設為0

若回傳值加總為40代表四個點都回傳投影片中的 r

若回傳值小於40代表其中有r有q可能也有s，所以將他mod10就可以取出q

```
def yokoi(img):  
    size = img.shape[0]  
    ans = np.zeros((size, size), dtype=int)  
    for i in range(size):  
        for j in range(size):  
            if img[i][j]==1:  
                a1 = h_func(img, [i,j], [i,j+1], [i-1,j+1], [i-1,j])  
                a2 = h_func(img, [i,j], [i-1,j], [i-1,j-1], [i,j-1])  
                a3 = h_func(img, [i,j], [i,j-1], [i+1,j-1], [i+1,j])  
                a4 = h_func(img, [i,j], [i+1,j], [i+1,j+1], [i,j+1])  
                a = a1+a2+a3+a4  
                ans[i][j]=5 if a==40 else a%10  
    return ans
```

## 5. h\_func 傳入四個對應的pixel值

判斷x2有沒有超出邊界，超出邊界代表x1不可能等於x2，所以為s回傳0

若沒超出邊界再來判斷x1是否等於x2，若不等於一樣為s回傳0

若x1=x2則來觀察x3 x4與x1的關係

如x3超出邊界則x4也會超出邊界，且x3 x4不會等於x1，所以為q回傳1

如x3 x4沒超出邊界，則判斷x3 x4是否都與x1相同，若相同為r回傳10，若不同則為q，回傳1

```
def h_func(img, x1, x2, x3, x4):
    size = img.shape[0]
    if x2[0]>=size or x2[1]>=size or x2[0]<0 or x2[1]<0:
        return 0
    elif (img[x1[0]][x1[1]]) != (img[x2[0]][x2[1]]):
        return 0
    if x3[0]>=size or x3[1]>=size or x3[0]<0 or x3[1]<0:
        return 1
    elif img[x3[0]][x3[1]]==img[x1[0]][x1[1]] and img[x4[0]][x4[1]]==img[x1[0]][x1[1]]:
        return 10
    else:
        return 1
```

結果：

