Homework 6

- 使用之前寫好的binarize函數對lena.bmp進行binarize

```
def binarize(img, threshold):
    for i in range(512):
        for j in range(512):
            if int(img[i][j][0]) >= threshold:
                img[i][j] = [255, 255, 255]
            else:
                img[i][j] = [0, 0, 0]
    return img
```

```
# open and binarize lena.bmp
img = cv2.imread('lena.bmp')
img = binarize(img, 128)
```

```
# downsample lena.bmp
img_down = np.zeros((64, 64))
for i in range(img_down.shape[0]):
    for j in range(img_down.shape[1]):
        img_down[i][j] = img[8 * i][8 * j][0]
```

- 先開啟一個64x64的矩陣img_down，用來裝downsize後的value
- 以8x8為單位，將topmost-left pixel設為downsampled data寫入img_down中

```
# function for Yokoi Connectivity Number
def h(b, c, d, e):
    if b == c and (d != b or e != b):
        return 'q'
    if b == c and (d == b and e == b):
        return 'r'
    return 's'
```

- 先將Yokoi Connectivity Number的計算公式寫成函數，以利接下來計算整張圖

```
for i in range(img_down.shape[0]):
    for j in range(img_down.shape[1]):
        if img_down[i][j] > 0:
            if i == 0:
                if j == 0:
                    # top-left
                    x7, x2, x6 = 0, 0, 0
                    x3, x0, x1 = 0, img_down[i][j], img_down[i][j + 1]
                    x8, x4, x5 = 0, img_down[i + 1][j], img_down[i + 1][j + 1]

                elif j == img_down.shape[1] - 1:
                    # top-right
                    x7, x2, x6 = 0, 0, 0
                    x3, x0, x1 = img_down[i][j - 1], img_down[i][j], 0
                    x8, x4, x5 = img_down[i + 1][j - 1], img_down[i + 1][j], 0
                else:
                    # top-row
                    x7, x2, x6 = 0, 0, 0
                    x3, x0, x1 = img_down[i][j - 1], img_down[i][j], img_down[i][j + 1]
                    x8, x4, x5 = img_down[i + 1][j - 1], img_down[i + 1][j], img_down[i + 1][j + 1]
            elif i == img_down.shape[0] - 1:
                if j == 0:
                    # bottom-left
                    x7, x2, x6 = 0, img_down[i - 1][j], img_down[i - 1][j + 1]
                    x3, x0, x1 = 0, img_down[i][j], img_down[i][j + 1]
                    x8, x4, x5 = 0, 0, 0
                elif j == img_down.shape[1] - 1:
                    # bottom-right
                    x7, x2, x6 = img_down[i - 1][j - 1], img_down[i - 1][j], 0
                    x3, x0, x1 = img_down[i][j - 1], img_down[i][j], 0
                    x8, x4, x5 = 0, 0, 0
                else:
                    # bottom-row
                    x7, x2, x6 = img_down[i - 1][j - 1], img_down[i - 1][j], img_down[i - 1][j + 1]
                    x3, x0, x1 = img_down[i][j - 1], img_down[i][j], img_down[i][j + 1]
                    x8, x4, x5 = 0, 0, 0
            else:
                if j == 0:
                    x7, x2, x6 = 0, img_down[i - 1][j], img_down[i - 1][j + 1]
                    x3, x0, x1 = 0, img_down[i][j], img_down[i][j + 1]
                    x8, x4, x5 = 0, img_down[i + 1][j], img_down[i + 1][j + 1]
                elif j == img_down.shape[1] - 1:
                    x7, x2, x6 = img_down[i - 1][j - 1], img_down[i - 1][j], 0
                    x3, x0, x1 = img_down[i][j - 1], img_down[i][j], 0
                    x8, x4, x5 = img_down[i + 1][j - 1], img_down[i + 1][j], 0
                else:
                    x7, x2, x6 = img_down[i - 1][j - 1], img_down[i - 1][j], img_down[i - 1][j + 1]
                    x3, x0, x1 = img_down[i][j - 1], img_down[i][j], img_down[i][j + 1]
                    x8, x4, x5 = img_down[i + 1][j - 1], img_down[i + 1][j], img_down[i + 1][j + 1]
            a1 = h(x0, x1, x6, x2)
            a2 = h(x0, x2, x7, x3)
            a3 = h(x0, x3, x8, x4)
            a4 = h(x0, x4, x5, x1)

            if a1 == 'r' and a2 == 'r' and a3 == 'r' and a4 == 'r':
                ans = 5
            else:
                ans = 0
                for a_i in [a1, a2, a3, a4]:
                    if a_i == 'q':
                        ans += 1

            print('%d' %ans,end=' ')

        else:
            print(' ', end=' ')

        if j == img_down.shape[1] - 1:
            print('')
```

- 根據上課簡報中的方法再加上事先寫好的函數，可以計算出a1、a2、a3、a4的類別，計算出每個pixel的Yokoi Connectivity Number，最後將每個pixel計算出來的Yokoi Connectivity Number印出，得到整張圖的結果如下：

```
1 1 1 1 1 1 1 1                 1 2 1 1 1 1 1 1 1 1 1 1 2 2 3 2 2 2 2 1           1 1 1 1 1 1 1 1 1 1 1 1               0     0
1 5 5 5 5 5 5 1                   1 1 5 5 5 5 5 5 5 5 1 1   2   1 1     1 1       1 1 5 5 5 5 5 5 5 5 5 1 1                 0
1 5 5 5 5 5 5 1                 1   2 1 1 5 5 5 5 1 1 2     2 1 1 1 2 2 2 1         1 5 5 5 5 5 5 5 5 5 5 1             2 1
1 5 5 5 5 5 5 1                 1   2   1 5 5 1 1 2   2 2 2 2 1 5 1 1               1 5 5 5 5 5 5 5 5 5 5 1 1           1
1 5 5 5 5 5 5 1                   2 2   2 1 1 2   2 2         1 2 1   0   0         1 5 5 5 5 5 5 5 5 5 5 5 1 1       0
1 5 5 5 5 5 5 1                   1     2     2 1   2           1       1           1 5 5 5 5 5 5 5 5 5 5 5 5 1     0
1 5 5 5 5 5 5 1                       1 2   1     1 2 1 1 1 1         1 3 2 1       1 5 5 5 5 5 5 5 5 5 5 5 5 1 1
1 5 1 1 1 5 5 1                       1 3 2 2   1 1 5 5 5 5 1 1 1 1                 1 5 5 5 5 5 5 5 5 5 5 5 5 5 1
1 1 1   1 5 5 1                         1     1 2 1 5 5 5 5 5 5 5 1 1               1 5 5 5 5 5 5 5 5 5 5 5 5 1 1
1 1     1 5 5 1                                   2 1 1 5 5 5 5 5 5 1 1             1 5 5 1 1 1 5 5 5 5 5 5 5 1 1
2 1     1 5 5 1                                   2   1 5 5 5 5 5 5 5 1 1 1         1 5 5 1   1 1 5 5 5 5 5 1 1
1       1 5 5 1                                   2   1 5 5 5 5 5 5 5 5 5 1 1       1 5 5 1     1 1 5 5 5 1                   1
        1 5 5 1                               1 1 2 1 1 5 5 5 5 5 5 5 5 5 5 1       1 5 5 1       1 5 5 1 1                 1 2
        1 5 5 1                               1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1     1 5 5 1       1 1 1 1                 1 1 1
        1 5 5 1                 1           2 2 2 1 1 5 5 5 5 5 5 5 5 5 5 5 5 1 1   1 1 5 1         1 1                 1 1 5 1
        1 5 5 1                 2         2 2   1   1 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1   1 5 1     1 1 1 1 1               1 5 5 1
        1 5 5 1                 2         1       1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1   1 5 1   1 1 5 5 5 1             1 1 5 5 1
        1 5 5 1                 2               1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1 1 5 1 1 1 5 5 5 1 1           1 1 5 5 5 1
        1 5 5 1               1 2             1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1             1 5 5 5 5 1
        1 5 5 1               1 1       0   2 2 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1 2           1 1 5 5 5 5 1
        1 5 5 1               1 1 1       2 2   1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1   1           1 5 5 5 5 5 1
        1 5 5 1               1 5 1 1     1   1 2 5 1 1 2 1 1 1 1 1 2 1 1 1 5 5 5 5 5 5 5 5 5 1 1 1             1 1 5 5 5 5 5 1
        1 5 5 1               1 5 5 2 1     1   1 2 1   1   1 1     1     1 5 5 5 5 5 5 5 1 1 1     0           1 5 5 5 5 5 5 1
        1 5 5 1               1 1 5 1     1 3 2   2                     1 1 5 5 5 5 5 1 1 1       0           1 1 5 5 5 5 5 5 1
        1 5 5 1                 1 5 1   0     3 2 2                   1 1 5 5 5 5 1 1 1     1 2 1             1 5 5 5 5 5 5 5 1
        1 5 5 1                 1 2 2 1       2                       1 5 5 5 5 5 1       1 3 1             1 1 5 5 5 5 5 5 5 1
        1 5 5 1                   2     0     1                     1 1 5 5 5 5 5 1 1       1               1 1 5 5 5 5 5 5 5 1
        1 5 5 1                   2       0           0         1 1 5 5 5 5 5 5 5 1     0               1   1 5 5 5 5 5 5 5 1
        1 5 5 1                   2                           1 1 5 5 5 5 5 5 5 5 1                     2 1 1 5 5 5 5 5 5 5 1
        1 5 5 1                   1     0                   1 1 5 5 5 5 5 5 5 5 5 1                     1 5 5 5 5 5 5 5 5 5 1
        1 5 5 1                     1                     1 1 5 1 1 1 1 5 5 5 5 5 2 1     1           1 1 5 5 5 5 5 5 5 5 5 1
        1 5 5 1                 1   1                   1 1 1 1 1       1 1 5 5 5 1 1       2           1 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1               1 3 1                     1 1 1           1 5 1 1 1         2           1 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1             1 2 1   0                 1 1 2 1       1     1 1 1     1       2         1 1 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1             1 1                       1 1 1   1     2 2 1   1 1     1       2         1 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1         1 2     0         1           2 1   1 2 1     1 1   1 1 1 1         2         1 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1           1           1 2         2 2     1 5 1 1 1 1 1 1 1 5 5 1         2       1 1 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1       1                             2       1 5 5 5 5 5 1 1 1 5 5 1 1       1       1 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1       2         0           0     2 2     1 2 5 5 5 5 5 1   1 5 5 5 1         1     1 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1       1                           1         1 5 5 5 5 1 1   1 1 5 1 1         2   1 1 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1             0   0             2 1           1 5 5 5 5 1   1   1 5 1           2   1 5 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1                               2             1 5 5 5 5 1 1 2   1 5 1           2   1 5 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1                   1       1   1           1 1 5 5 5 5 5 5 1 1 1 1 1           2   1 5 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1                   2     2 2               1 1 1 5 1 1 1 1 1 2 1 2             2 1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 1
        1 5 5 1     0             1   1 2                     1 5 1         2   1             1 5 5 5 5 5 5 5 1 1 1 5 5 5 5 5 1
        1 5 5 1               0     0     0                   1 1 1 1     1 2 1               1 5 5 5 5 5 5 5 1   1 5 5 5 5 5 1
        1 5 5 1                         0                       1 1 1 1 1 1 1 1               1 5 5 5 5 5 5 5 1   1 5 5 5 5 5 1
        1 5 5 1                 0                                 1 1 5 5 5 1                 1 5 5 5 5 5 5 5 1   1 5 5 5 5 1 1
        1 5 5 1                                                     1 5 5 5 1                 2 1 1 1 1 1 1 1 1   1 5 5 5 1 1
        1 1 5 2 1             1       1 2                     1 2 2 1 5 5 5 1 1               2           1 1   1 1 5 5 1 1
1         1 5 1   0           1         1                         1 5 5 5 5 5 1 1 1           2 1 1 1           1 5 5 1 1
2 2       1 5 1 1                     1                           1 5 5 5 5 5 5 5 1 1 1       1 5 5 1 1 1       1 5 1 1
  2 2     1 5 1 1                     1                           1 5 5 5 5 5 5 5 5 5 1       1 5 5 5 5 1     1 1 5 1
    2     1 5 1                         0   1                 1 1 1 5 5 5 5 5 5 5 5 5 1 1     1 5 5 5 1 1     1 5 1 1
    2     1 5 2 1         0                 1                 1 5 5 5 5 5 5 5 5 5 5 5 5 1 1   1 5 5 5 1   1 2 1 5 1
    2     1 5 1                       1 2 1                   1 5 5 5 5 5 5 5 5 5 5 5 5 5 1   1 5 5 5 1 1   1 5 5 1
    2     1 5 1 1                                       0     1 5 5 5 5 5 5 5 5 5 5 5 5 5 1   1 1 5 5 5 1   1 5 1 1
    2 1   1 5 1 1                           1 1               1 5 5 5 5 5 5 5 5 5 5 5 5 5 1     1 1 1 1 1 1 1 5 1
    1 1   1 5 1                       0                     1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1         1 1 1 5 1 1
    1 1   1 5 1                                             1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1             1 5 1
    1 1   1 5 1                         0                 1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1             2 1 1
    1 1   1 5 1                                           1 1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1 1           1
    1 1   1 5 1                                         0   1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 1
    1 1   1 1 1                 0                         1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```