

首先先將原圖 512*512 的大小，每 8*8 的區塊掃過整張圖，取每一塊的最左上角的 pixel，並輸出成新的 64*64 的 binary image，接著對 binary image 做 Yokoi 的演算法。

以每個 pixel 為中心設為 x_0 ，周圍 8-connect 的 pixel 編號如下：

8-connected neighborhood

x_7	x_2	x_6
x_3	x_0	x_1
x_8	x_4	x_5

$$\text{跑 } h(b, c, d, e) = \begin{cases} q & \text{if } b = c \text{ and } (d \neq b \vee e \neq b) \\ r & \text{if } b = c \text{ and } (d = b \wedge e = b) \\ s & \text{if } b \neq c \text{ and } (d = b \wedge e = b) \end{cases}$$

	x_2	x_6
	x_0	x_1

x_7	x_2	
x_3	x_0	

x_3	x_0	
x_8	x_4	

	x_0	x_1
	x_4	x_5

照著上述 4 個順序跑 : $a_1 = h(x_0, x_1, x_6, x_2)$

$$a_2 = h(x_0, x_2, x_7, x_3)$$

$$a_3 = h(x_0, x_3, x_8, x_4)$$

$$a_4 = h(x_0, x_4, x_5, x_1)$$

$$f(a_1, a_2, a_3, a_4) = \begin{cases} 5 & \text{if } a_1 = a_2 = a_3 = a_4 = r \\ n & \text{where } n = \text{number of } \{a_k | a_k = q\}, \text{ otherwise} \end{cases}$$

再跑 $f(a_1, a_2, a_3, a_4)$ 即可

- Lena 64x64 binary 圖



●

[illegible]

● yokoi_num_without_0



- 程式碼

```
def binary_image(coulmn,row,pix,lena):
    pix=lena.load()
    img_new=Image.new(lena.mode,(66,66))
    for i in range(0,coulmn,8):
        for j in range(0,row,8):
            if pix[i,j] < 128:
                img_new.putpixel((int(i/8)+1 ,int(j/8)+1),0)
            else:
                img_new.putpixel((int(i/8)+1 ,int(j/8)+1),255);
    img_new.save('lena_binary64x64.bmp')
    array=np.array(img_new)
    return array
```

```
def h(b,c,d,e):
    if b==c and ((d != b)or(e != b)):
        return 'q'
    if b==c and ((d == b)or(e == b)):
        return 'r'
    if b!=c:
        return 's'
```

```
def f(a1,a2,a3,a4):
    if a1==a2==a3==a4=='r':
        return 5
    else:
        num=0
        if a1=='q':
            num += 1
        if a2=='q':
            num += 1
        if a3=='q':
            num += 1
        if a4=='q':
            num += 1
        return num
```

```
def Yokoi(matrix):
```

```

yokoi_matrix = np.zeros((64,64),dtype=int)
for i in range(1,65):
    for j in range(1,65):
        if matrix[i,j]==255:
            x0 = matrix[i][j]
            x1 = matrix[i][j+1]
            x2 = matrix[i-1][j]
            x3 = matrix[i][j-1]
            x4 = matrix[i+1][j]
            x5 = matrix[i+1][j+1]
            x6 = matrix[i-1][j+1]
            x7 = matrix[i-1][j-1]
            x8 = matrix[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)

            yokoi_matrix[i-1][j-1] = f(a1,a2,a3,a4)
    return yokoi_matrix

```

```

lena=Image.open("lena.bmp")
pix=lena.load()
coulmn,row=lena.size
matrix=binary_image(coulmn,row,pix,lena)
array=Yokoi(matrix)
f = open('yokoi_num.txt', 'w')
k = open('yokoi_num_without_0.txt', 'w')
for i in range(64):
    for j in range(64):
        f.write("%d " %(array[i][j]))
    f.write("\n")
for i in range(64):
    for j in range(64):
        if array[i][j]==0:
            k.write(' ')
        else:

```

```
        k.write("%d " %(array[i][j]))  
    k.write("\n")  
f.close()  
k.close()
```