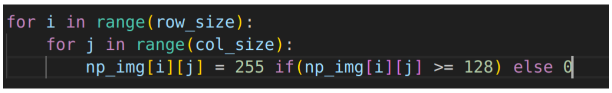
Computer Vision HW6

R12922054 資工所 邱信瑋

1. Write a program which counts the Yokoi connectivity number on a downsampled image(lena.bmp).

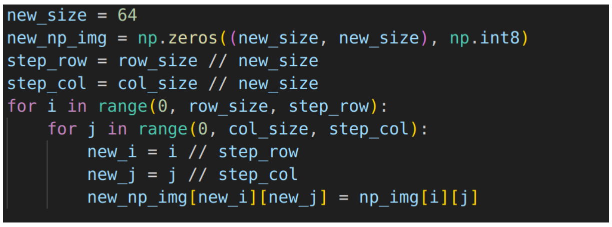
How to implement :

1. step 1: Binarize the benchmark image lena as in HW2.



```
for i in range(row_size):
    for j in range(col_size):
        np_img[i][j] = 255 if(np_img[i][j] >= 128) else 0
```

2. step 2: Downsampling Lena from 512x512 to 64x64.



```
new_size = 64
new_np_img = np.zeros((new_size, new_size), np.int8)
step_row = row_size // new_size
step_col = col_size // new_size
for i in range(0, row_size, step_row):
    for j in range(0, col_size, step_col):
        new_i = i // step_row
        new_j = j // step_col
        new_np_img[new_i][new_j] = np_img[i][j]
```

3. step 3: Establish function H and F according to the PPT CH6, Figure 1 Figure 3.

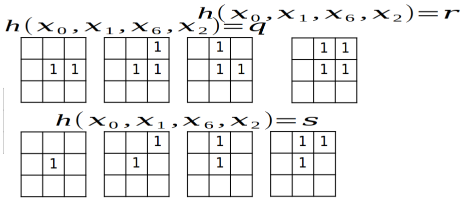
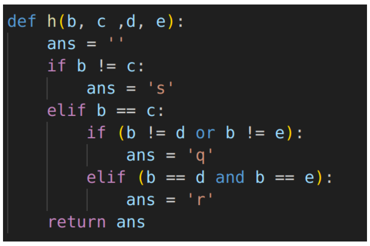


Figure 1: function_h_ref.



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

Figure 2: function_h_code.

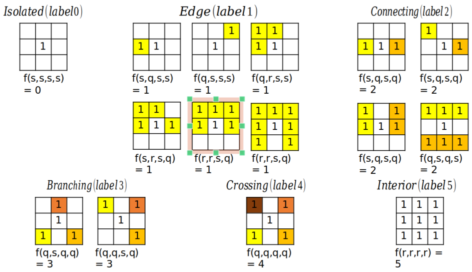
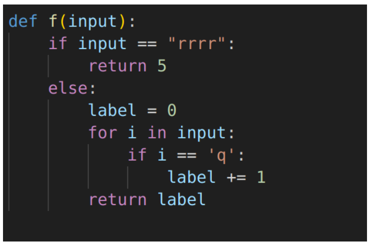


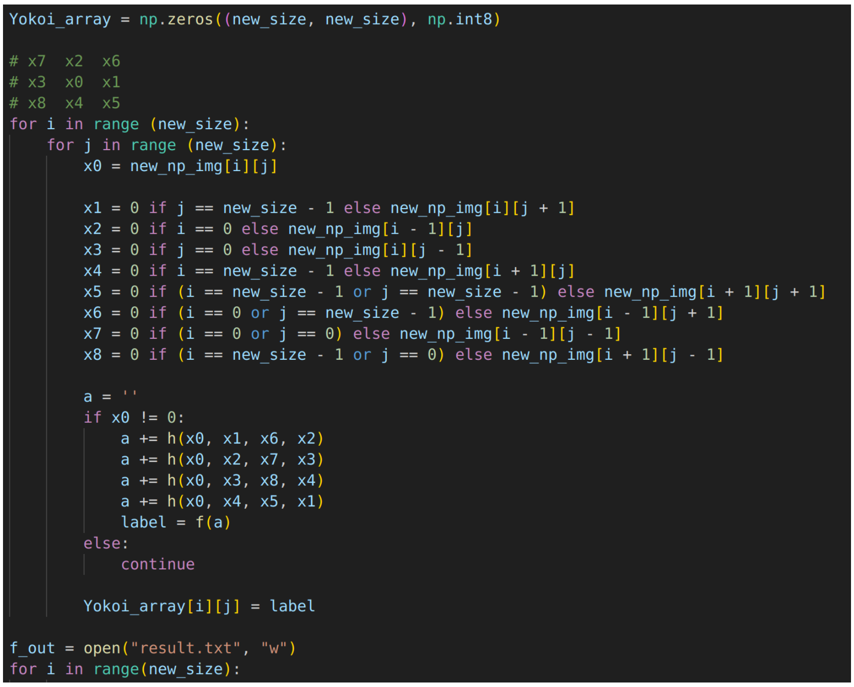
Figure 3: function_f_ref.



```python
def f(input):
    if input == "rrrr":
        return 5
    else:
        label = 0
        for i in input:
            if i == 'q':
                label += 1
        return label
```

Figure 4: function_f_code.

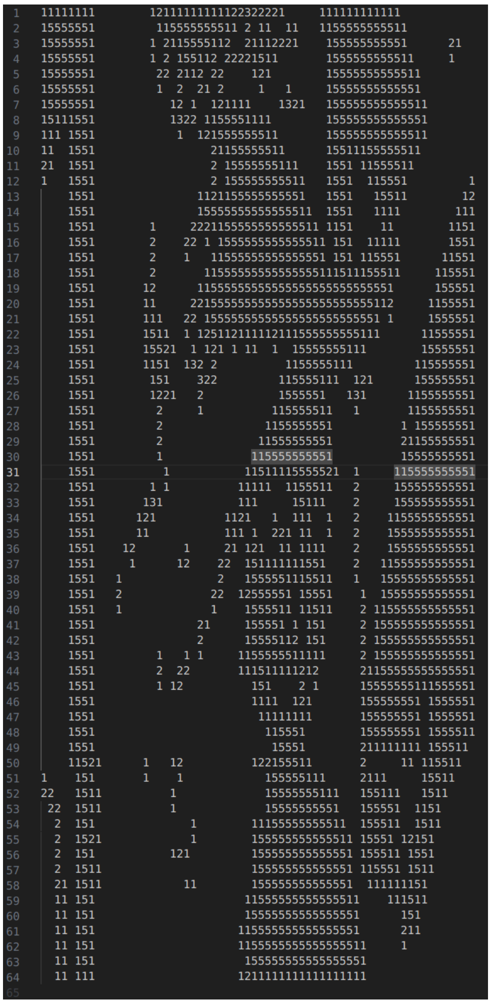
4. step 4: Caculate Yokoi label.



```python
Yokoi_array = np.zeros((new_size, new_size), np.int8)

# x7  x2  x6
# x3  x0  x1
# x8  x4  x5
for i in range (new_size):
    for j in range (new_size):
        x0 = new_np_img[i][j]

        x1 = 0 if j == new_size - 1 else new_np_img[i][j + 1]
        x2 = 0 if i == 0 else new_np_img[i - 1][j]
        x3 = 0 if j == 0 else new_np_img[i][j - 1]
        x4 = 0 if i == new_size - 1 else new_np_img[i + 1][j]
        x5 = 0 if (i == new_size - 1 or j == new_size - 1) else new_np_img[i + 1][j + 1]
        x6 = 0 if (i == 0 or j == new_size - 1) else new_np_img[i - 1][j + 1]
        x7 = 0 if (i == 0 or j == 0) else new_np_img[i - 1][j - 1]
        x8 = 0 if (i == new_size - 1 or j == 0) else new_np_img[i + 1][j - 1]

        a = ''
        if x0 != 0:
            a += h(x0, x1, x6, x2)
            a += h(x0, x2, x7, x3)
            a += h(x0, x3, x8, x4)
            a += h(x0, x4, x5, x1)
            label = f(a)
        else:
            continue

        Yokoi_array[i][j] = label

f_out = open("result.txt", "w")
for i in range(new_size):
```

Figure 5: caculate_Yokoi_code.

5. step 5: Get the result of Yokoi label and output it to .txt file.



```
11111111        12111111111122322221     111111111111
15555551         115555555511 2 11  11   1155555555511
15555551        1 2115555112  21112221    155555555551      21
15555551        1 2 155112 22221511       1555555555511     1
15555551         22 2112 22    121        15555555555511
15555551         1  2  21 2     1   1     15555555555551
15555551           12 1  121111    1321   155555555555511
15111551           1322 1155551111        155555555555551
111 1551            1  121555555511       155555555555511
11  1551                 21155555511      15511155555511
21  1551                 2 15555555111    1551 11555511
1   1551                 2 155555555511   1551  115551         1
    1551               1121155555555551   1551   15511        12
    1551               15555555555555511  1551   1111        111
    1551        1     2221155555555555511 1151    11        1151
    1551        2    22 1 1555555555555511 151  11111       1551
    1551        2    1   11555555555555551 151 115551      11551
    1551        2       11555555555555555511511155511     115551
    1551       12      11555555555555555555555555551      155551
    1551       11     221555555555555555555555555112     1155551
    1551       111   22 15555555555555555555555551 1     1555551
    1551       1511  1 125112111112115555555555111      11555551
    1551       15521  1 121 1 11  1  15555555111        15555551
    1551       1151  132 2          1155555111         115555551
    1551        151    322         115555111  121      155555551
    1551        1221   2           1555551   131      1155555551
    1551         2     1          115555511   1       1155555551
    1551         2               1155555551          1 155555551
    1551         2              11555555551          21155555551
    1551         1             115555555551          15555555551
    1551          1           11511115555521  1     115555555551
    1551        1 1          11111  1155511   2     155555555551
    1551       131           111     15111    2     155555555551
    1551      121          1121   1  111  1   2    1155555555551
    1551      11           111 1  221 11  1   2    1555555555551
    1551    12       1     21 121  11 1111    2    1555555555551
    1551     1      12    22  151111111551    2   11555555555551
    1551   1              2   1555551115511   1   15555555555551
    1551   2             22  12555551 15551    1  15555555555551
    1551   1             1    1555511 11511    2 115555555555551
    1551               21     155551 1 151     2 155555555555551
    1551               2      15555112 151     2 155555555555551
    1551         1   1 1     115555511111      2 155555555555551
    1551         2  22       111511111212      21155555555555551
    1551         1 12          151    2 1      15555555111555551
    1551                       1111  121       155555551 1555551
    1551                        11111111       155555551 1555551
    1551                         115551        155555551 1555511
    1551                          15551        211111111 155511
    11521      1   12          122155511       2     11 115511
1    151       1    1            155555111     2111     15511
22   1511          1             15555555111   155111   1511
 22  1511          1             15555555551   155551  1151
  2  151              1        11155555555511  155511  1511
  2  1521             1        155555555555511 15551 12151
  2  151           121         155555555555551 155511 1551
  2  1511                      155555555555551 115551 1511
  21 1511            11        155555555555551  111111151
  11 151                      11555555555555511    111511
  11 151                      15555555555555551      151
  11 151                     115555555555555551      211
  11 151                     1155555555555555511     1
  11 151                      155555555555555551
  11 111                     1211111111111111111
```

Figure 6: Result-Yokoi_Label.