Report

Result:

First, binarize the image at 128.

To downsample the image from 512x512 to 64x64 composed of topmost-left pixels, from (0, 0), take a sample pixel every 8 columns and 8 rows.

To calculate Yokoi connectivity numbers for the image, first, pad the image with 0. Then, calculate Yokoi connectivity number for each pixel (r, c) by the following function.

```
def yokoi_number(r, c, img):
    a1 = h(img[r, c], img[r, c+1], img[r-1, c+1], img[r-1, c])
    a2 = h(img[r, c], img[r-1, c], img[r-1, c-1], img[r, c-1])
    a3 = h(img[r, c], img[r, c-1], img[r+1, c-1], img[r+1, c])
    a4 = h(img[r, c], img[r+1, c], img[r+1, c+1], img[r, c+1])

    if a1 == 'r' and a2 == 'r' and a3 == 'r' and a4 == 'r':
        return 5
    count = 0
    if a1 == 'q':
        count += 1
    if a2 == 'q':
        count += 1
    if a3 == 'q':
        count += 1
    if a4 == 'q':
        count += 1
    return count
```

```
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'
    if b != c:
        return 's'
```

Function `h(b, c, d, e)` is defined as in the lecture slides.

- for 4-connectivity

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

The function `yokoi_number(r, c, img)` calculate Yokoi connectivity number of the pixel (r, c). First, calculate a1, a2, a3, a4 by function `h(b, c, d, e)` as the following, where x0 = img(r, c) and the relative position as shown in the figure.

$$a_1 = h(x_0, x_1, x_6, x_2)$$
$$a_2 = h(x_0, x_2, x_7, x_3)$$
$$a_3 = h(x_0, x_3, x_8, x_4)$$
$$a_4 = h(x_0, x_4, x_5, x_1)$$

x_7	x_2	x_6
x_3	x_0	x_1
x_8	x_4	x_5

Then if a1 == a2 == a3 == a4 == ‘r’, return 5, else, return the number of ‘q’ in {a1, a2, a3, a4}