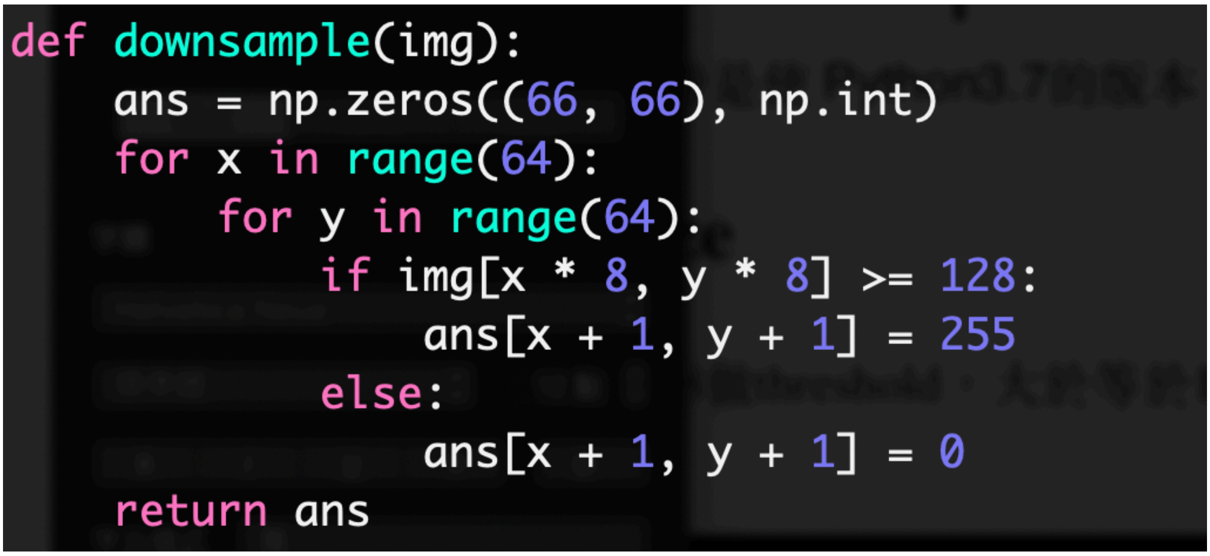
B06902042 劉愷爲

Computer Vision HW7

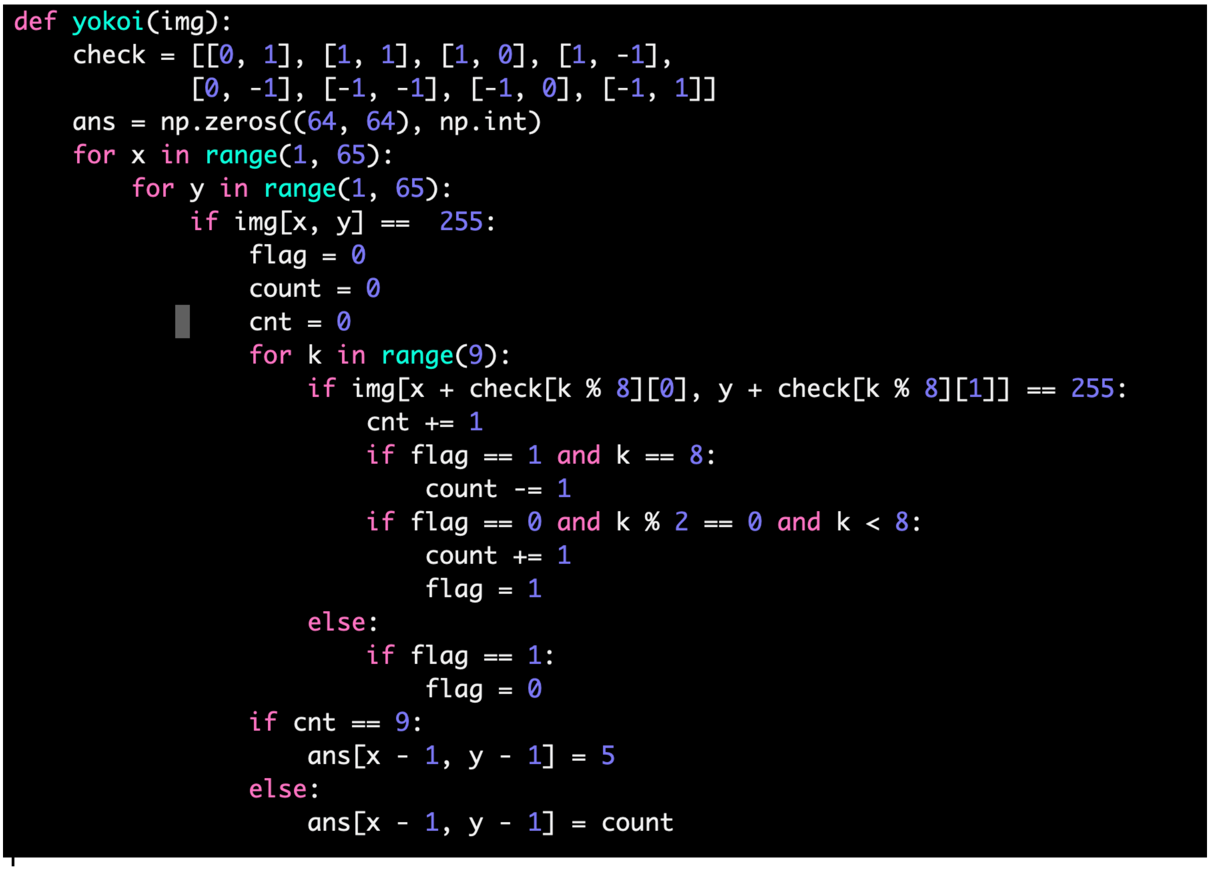
Part 1: Binarize + Downsampled

將512*512的圖以8*8的圖分割成64*64，並以每個部分最左上角的灰階值，作為64*64的灰階值。



```
def downsample(img):
    ans = np.zeros((66, 66), np.int)
    for x in range(64):
        for y in range(64):
            if img[x * 8, y * 8] >= 128:
                ans[x + 1, y + 1] = 255
            else:
                ans[x + 1, y + 1] = 0
    return ans
```

Part 2: Yokoi connectivity number(4-connected)

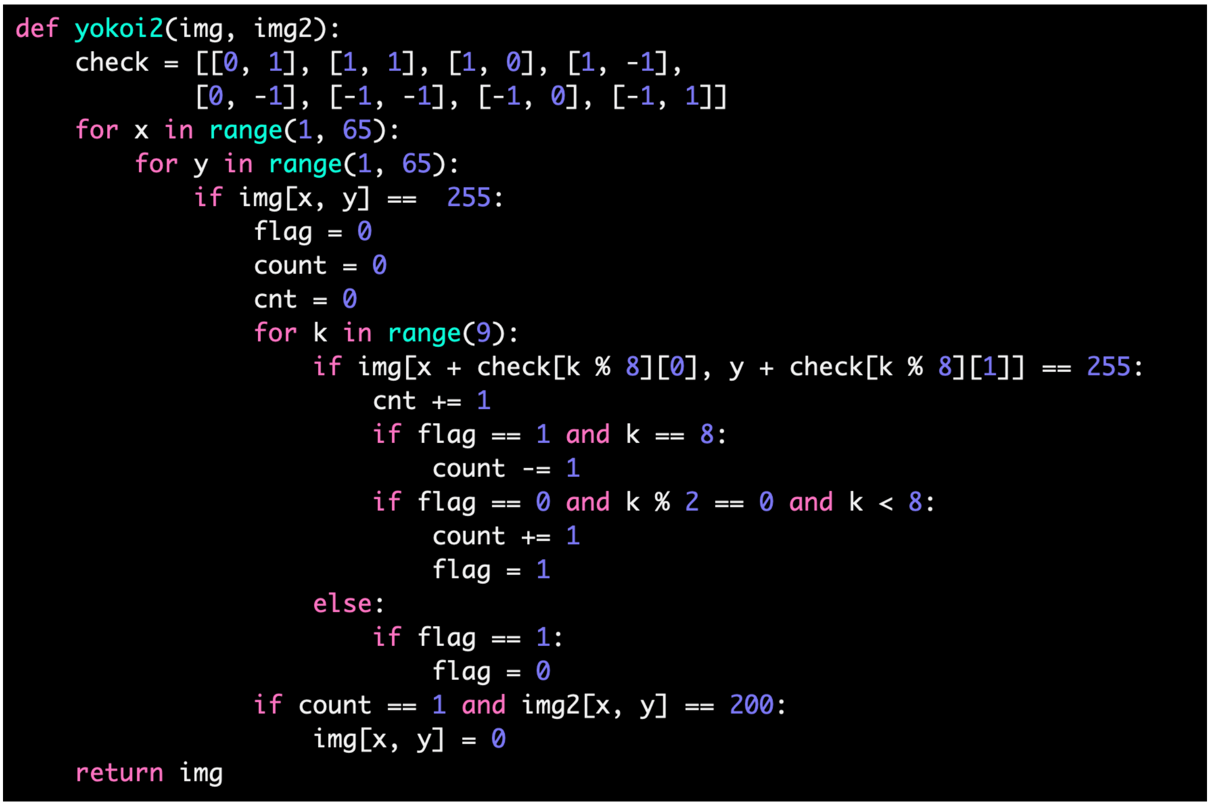


```
def yokoi(img):
    check = [[0, 1], [1, 1], [1, 0], [1, -1],
             [0, -1], [-1, -1], [-1, 0], [-1, 1]]
    ans = np.zeros((64, 64), np.int)
    for x in range(1, 65):
        for y in range(1, 65):
            if img[x, y] ==  255:
                flag = 0
                count = 0
                cnt = 0
                for k in range(9):
                    if img[x + check[k % 8][0], y + check[k % 8][1]] == 255:
                        cnt += 1
                        if flag == 1 and k == 8:
                            count -= 1
                        if flag == 0 and k % 2 == 0 and k < 8:
                            count += 1
                            flag = 1
                    else:
                        if flag == 1:
                            flag = 0
                if cnt == 9:
                    ans[x - 1, y - 1] = 5
                else:
                    ans[x - 1, y - 1] = count
```

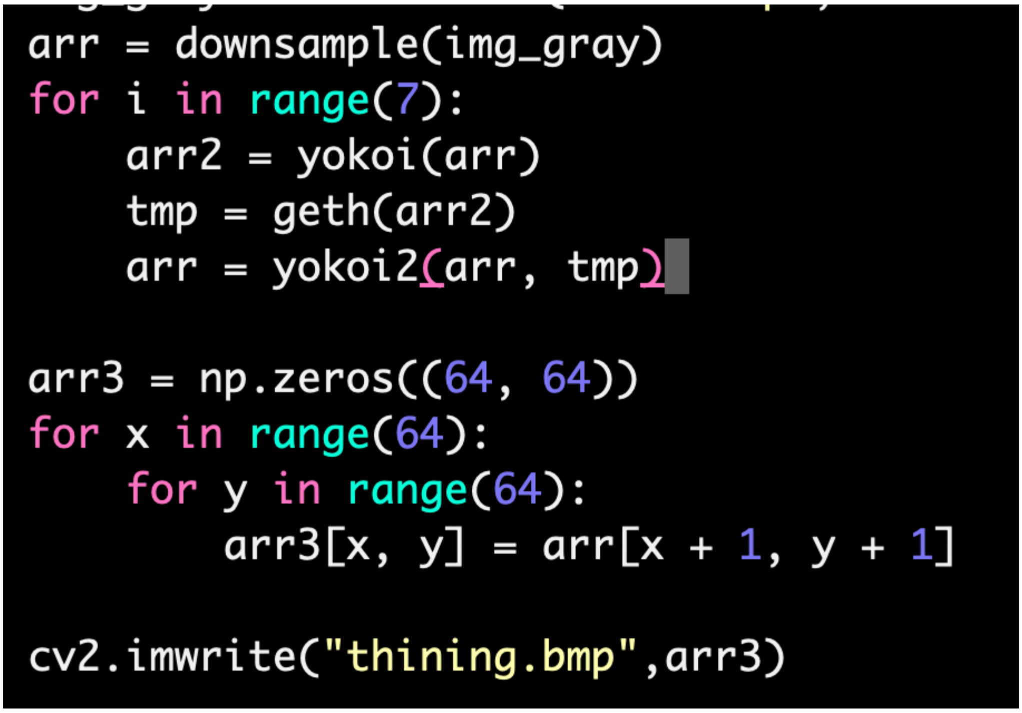
每次我直接從一個點繞一圈去找他的Yokoi Number，所以是從1~64。我在外面多加一個框框。因為我多加了一個框框，所以不用特別注意邊界問題。

Part 4:

後面的運算跟前面的Yokoi很像，只差在最後兩行的判斷 當count為1且img2為200時，把那一格設成0。整個流程做7次就會得到最終的圖



```
def yokoi2(img, img2):
    check = [[0, 1], [1, 1], [1, 0], [1, -1],
            [0, -1], [-1, -1], [-1, 0], [-1, 1]]
    for x in range(1, 65):
        for y in range(1, 65):
            if img[x, y] ==  255:
                flag = 0
                count = 0
                cnt = 0
                for k in range(9):
                    if img[x + check[k % 8][0], y + check[k % 8][1]] == 255:
                        cnt += 1
                        if flag == 1 and k == 8:
                            count -= 1
                        if flag == 0 and k % 2 == 0 and k < 8:
                            count += 1
                            flag = 1
                    else:
                        if flag == 1:
                            flag = 0
                if count == 1 and img2[x, y] == 200:
                    img[x, y] = 0
    return img
```



```
arr = downsample(img_gray)
for i in range(7):
    arr2 = yokoi(arr)
    tmp = geth(arr2)
    arr = yokoi2(arr, tmp)

arr3 = np.zeros((64, 64))
for x in range(64):
    for y in range(64):
        arr3[x, y] = arr[x + 1, y + 1]

cv2.imwrite("thining.bmp",arr3)
```

RESULT: