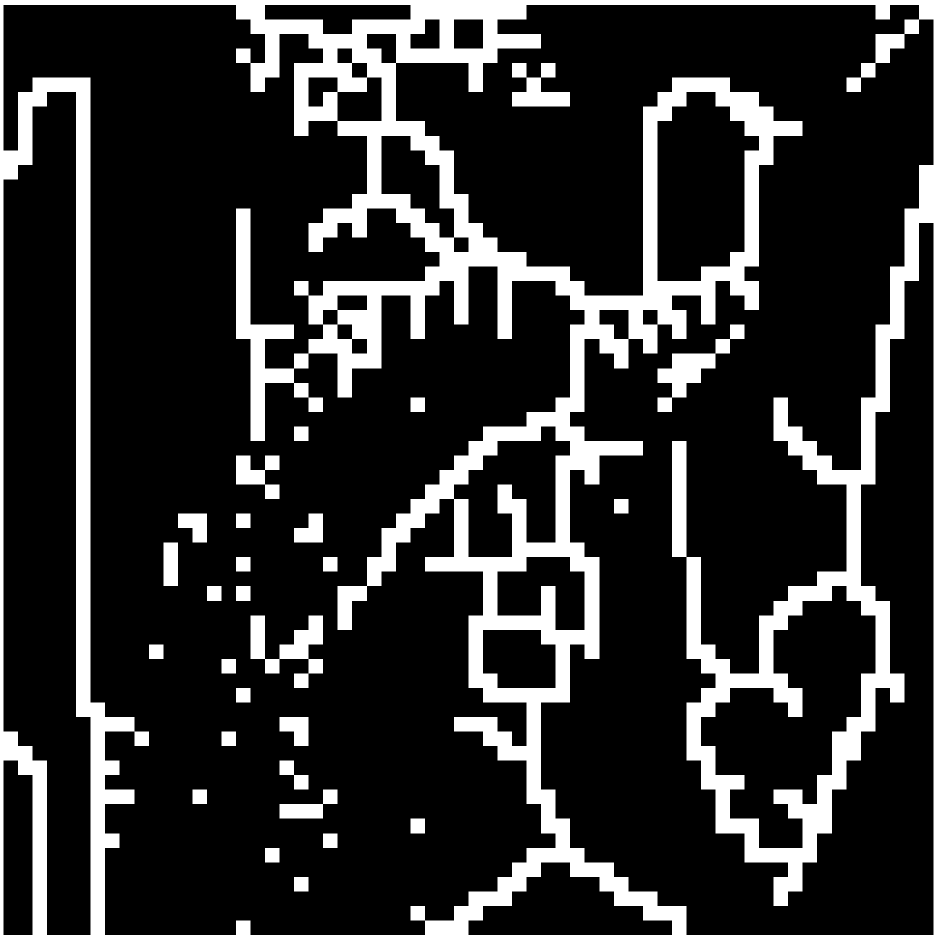
電腦視覺 HW7

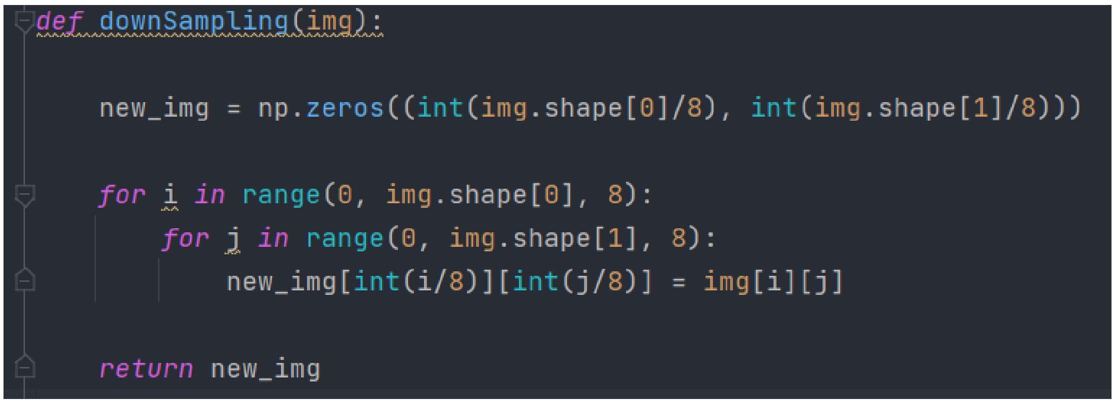
R10922176 陳冠潁

Thinning



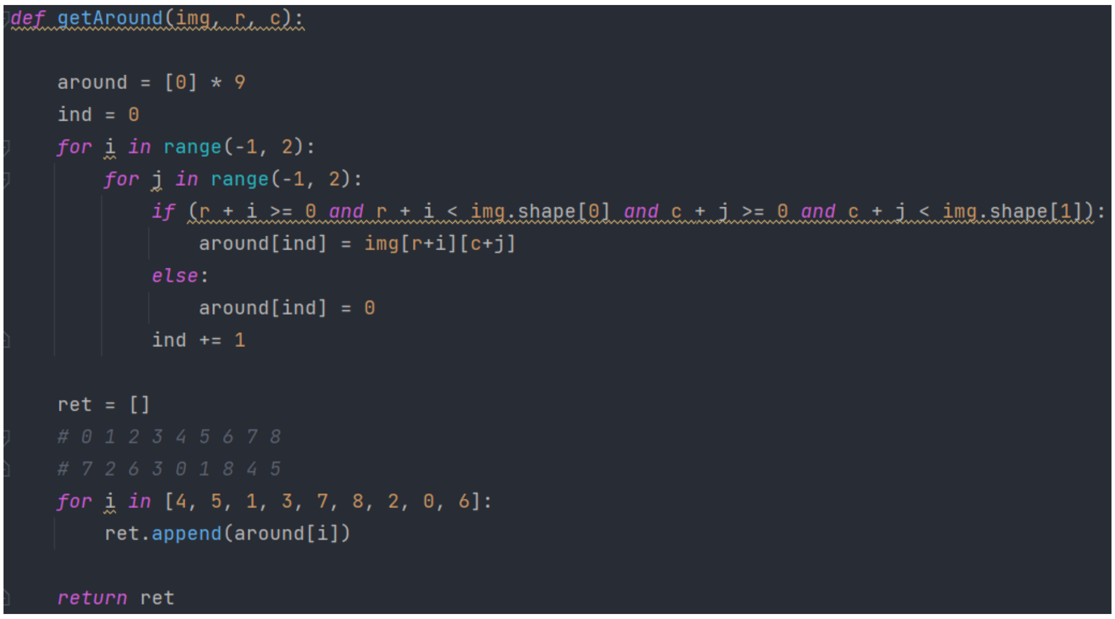
說明:

downSampling



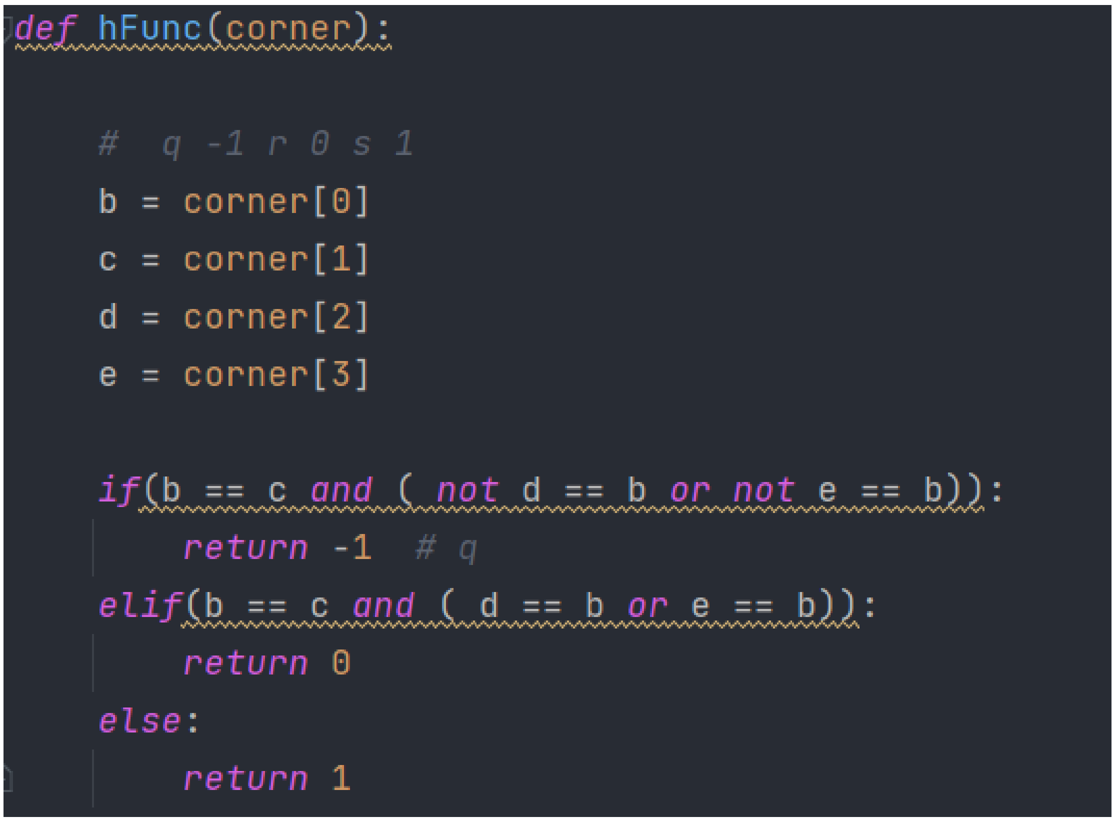
```python
def downSampling(img):

    new_img = np.zeros((int(img.shape[0]/8), int(img.shape[1]/8)))

    for i in range(0, img.shape[0], 8):
        for j in range(0, img.shape[1], 8):
            new_img[int(i/8)][int(j/8)] = img[i][j]

    return new_img
```

getAround 用來取得包含中心點周圍的像素值



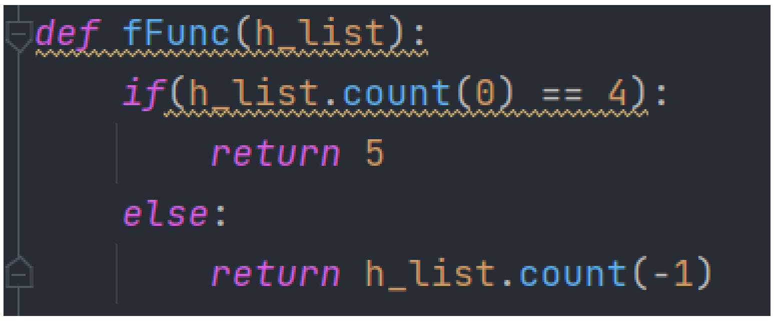
```python
def getAround(img, r, c):

    around = [0] * 9
    ind = 0
    for i in range(-1, 2):
        for j in range(-1, 2):
            if (r + i >= 0 and r + i < img.shape[0] and c + j >= 0 and c + j < img.shape[1]):
                around[ind] = img[r+i][c+j]
            else:
                around[ind] = 0
            ind += 1

    ret = []
    # 0 1 2 3 4 5 6 7 8
    # 7 2 6 3 0 1 8 4 5
    for i in [4, 5, 1, 3, 7, 8, 2, 0, 6]:
        ret.append(around[i])

    return ret
```

hfunc 照講義上的 h function 公式來判斷



```python
def hFunc(corner):

    #  q -1 r 0 s 1
    b = corner[0]
    c = corner[1]
    d = corner[2]
    e = corner[3]

    if(b == c and ( not d == b or not e == b)):
        return -1  # q
    elif(b == c and ( d == b or e == b)):
        return 0
    else:
        return 1
```

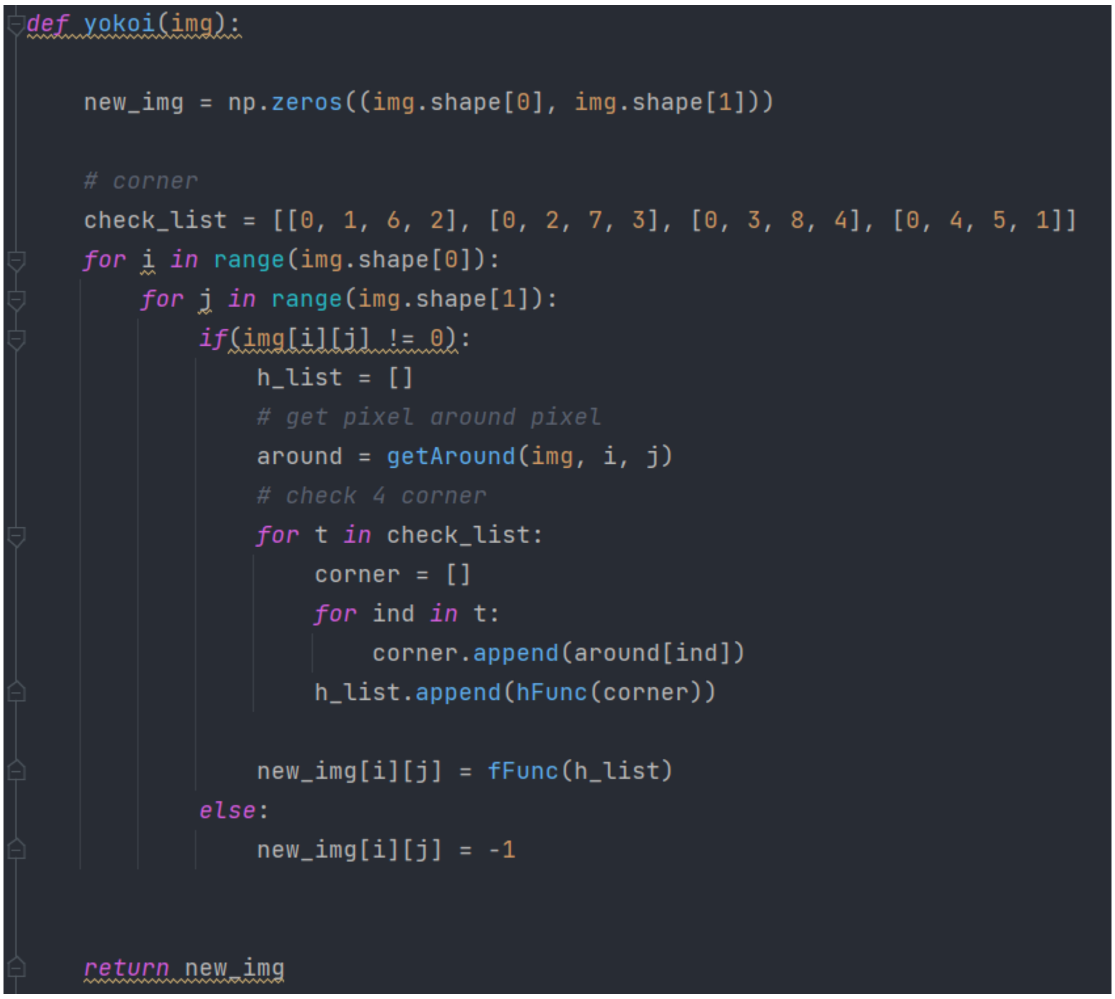
fFun 照講義上的 f function 判斷 q 的數量



```python
def fFunc(h_list):
    if(h_list.count(0) == 4):
        return 5
    else:
        return h_list.count(-1)
```

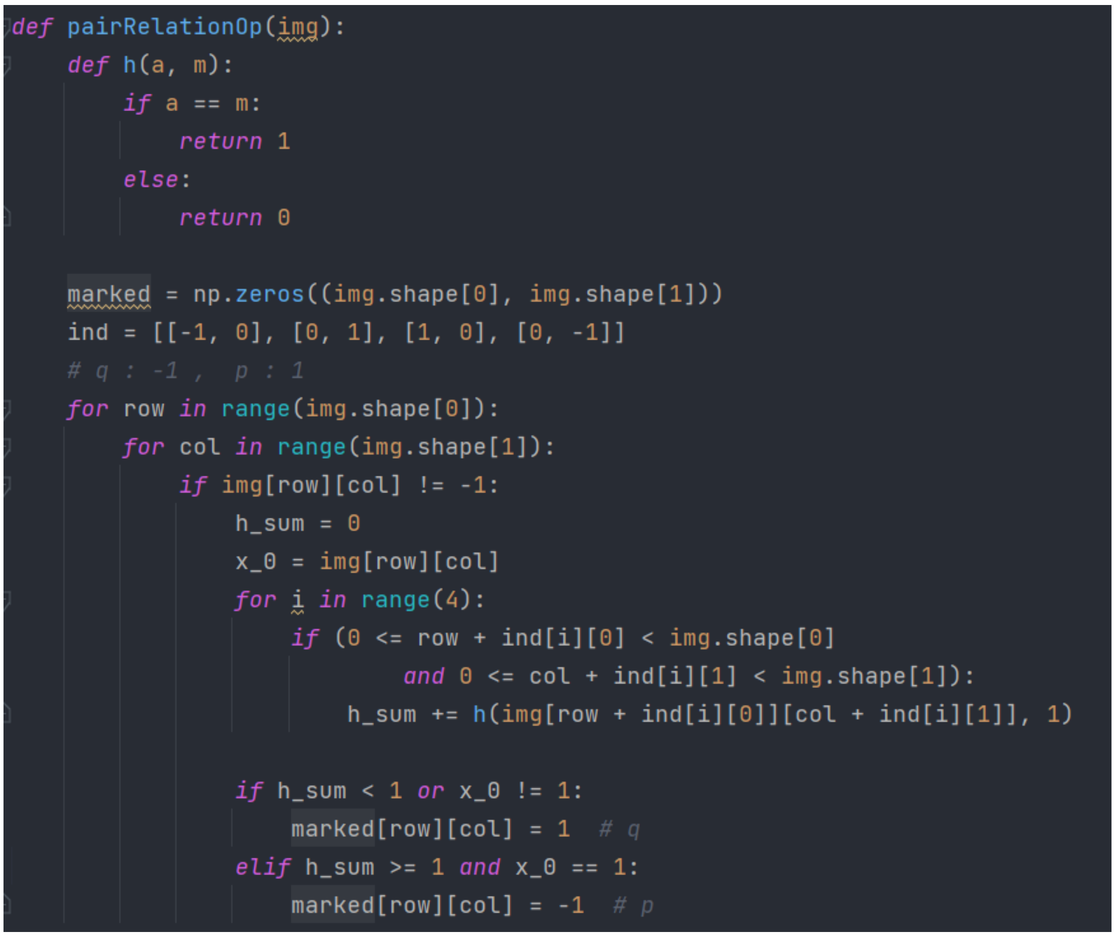
Yokoi:

對每個點先 getAround 之後將要用的 corner 點傳入 h function 做判斷，最後在將 4 個 corner 算完的結果傳入 f function 得到 yokoi number。



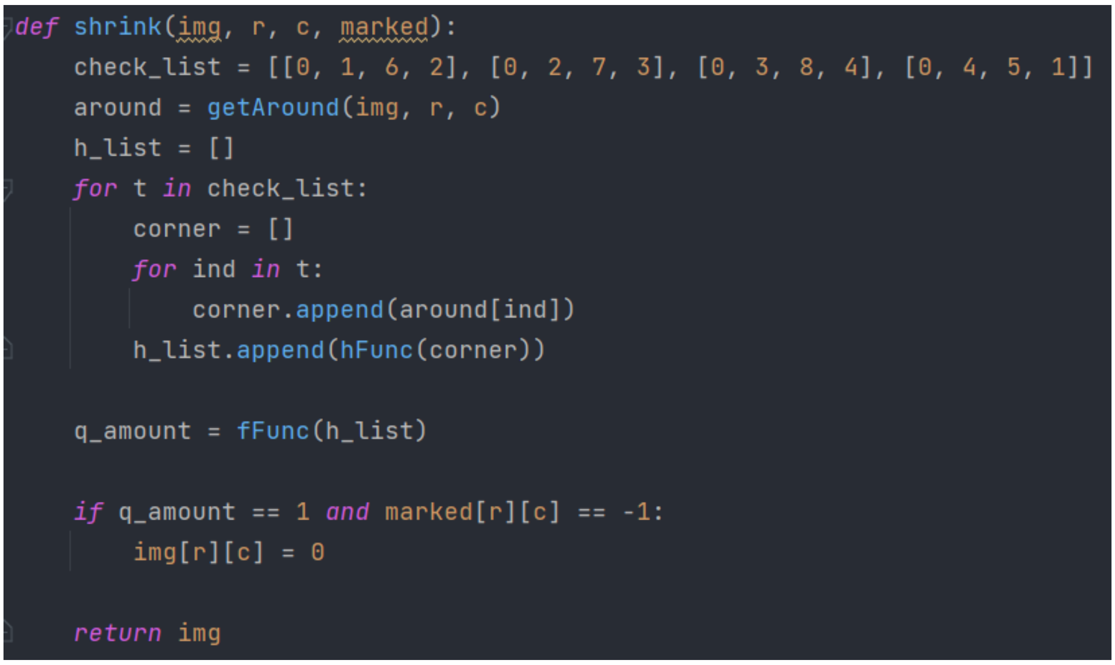
```python
def yokoi(img):

    new_img = np.zeros((img.shape[0], img.shape[1]))

    # corner
    check_list = [[0, 1, 6, 2], [0, 2, 7, 3], [0, 3, 8, 4], [0, 4, 5, 1]]
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            if(img[i][j] != 0):
                h_list = []
                # get pixel around pixel
                around = getAround(img, i, j)
                # check 4 corner
                for t in check_list:
                    corner = []
                    for ind in t:
                        corner.append(around[ind])
                    h_list.append(hFunc(corner))

                new_img[i][j] = fFunc(h_list)
            else:
                new_img[i][j] = -1


    return new_img
```

pairRelationOp 使用講義的公式 mark p 跟 q



```python
def pairRelationOp(img):
    def h(a, m):
        if a == m:
            return 1
        else:
            return 0

    marked = np.zeros((img.shape[0], img.shape[1]))
    ind = [[-1, 0], [0, 1], [1, 0], [0, -1]]
    # q : -1 ,  p : 1
    for row in range(img.shape[0]):
        for col in range(img.shape[1]):
            if img[row][col] != -1:
                h_sum = 0
                x_0 = img[row][col]
                for i in range(4):
                    if (0 <= row + ind[i][0] < img.shape[0]
                            and 0 <= col + ind[i][1] < img.shape[1]):
                        h_sum += h(img[row + ind[i][0]][col + ind[i][1]], 1)

                if h_sum < 1 or x_0 != 1:
                    marked[row][col] = 1  # q
                elif h_sum >= 1 and x_0 == 1:
                    marked[row][col] = -1  # p
```

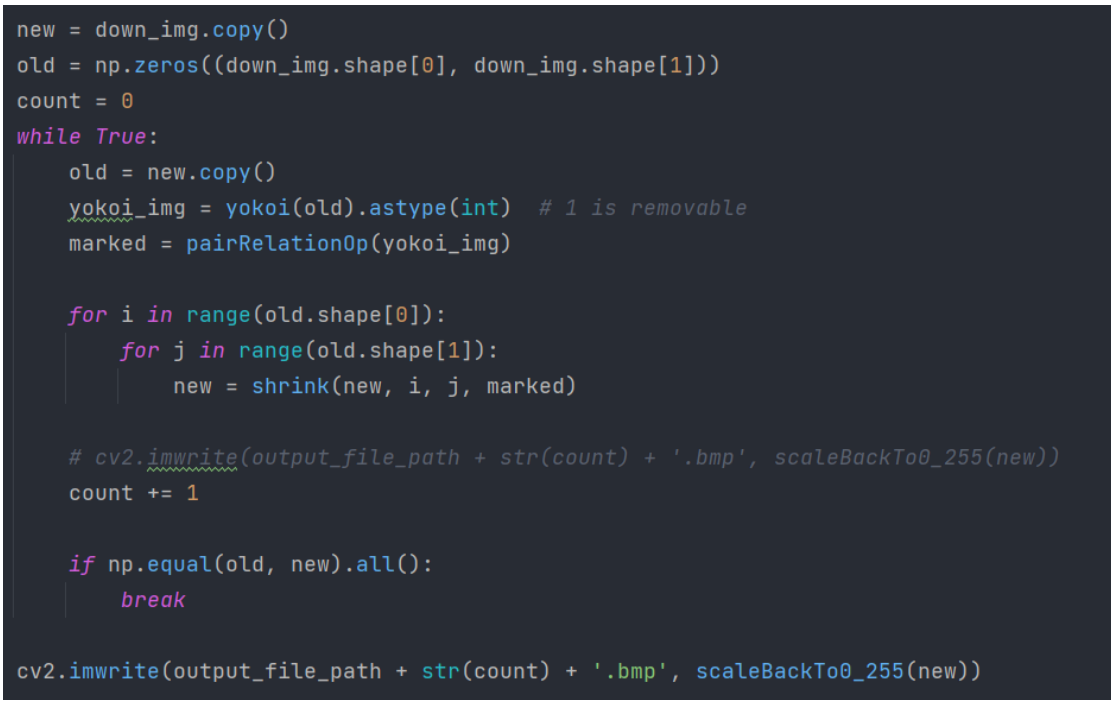
shrink 針對當前傳進來位於(r, c)的像素判斷是否要消除



```python
def shrink(img, r, c, marked):
    check_list = [[0, 1, 6, 2], [0, 2, 7, 3], [0, 3, 8, 4], [0, 4, 5, 1]]
    around = getAround(img, r, c)
    h_list = []
    for t in check_list:
        corner = []
        for ind in t:
            corner.append(around[ind])
        h_list.append(hFunc(corner))

    q_amount = fFunc(h_list)

    if q_amount == 1 and marked[r][c] == -1:
        img[r][c] = 0

    return img
```

Main

一次對一個 pixel 做 shrink 才能得到正確的結果，每一輪做完後跟上一輪比對，如果沒有改變了則停止。



```python
new = down_img.copy()
old = np.zeros((down_img.shape[0], down_img.shape[1]))
count = 0
while True:
    old = new.copy()
    yokoi_img = yokoi(old).astype(int)  # 1 is removable
    marked = pairRelationOp(yokoi_img)

    for i in range(old.shape[0]):
        for j in range(old.shape[1]):
            new = shrink(new, i, j, marked)

    # cv2.imwrite(output_file_path + str(count) + '.bmp', scaleBackTo0_255(new))
    count += 1

    if np.equal(old, new).all():
        break

cv2.imwrite(output_file_path + str(count) + '.bmp', scaleBackTo0_255(new))
```