

電腦視覺 HW6

R10922176 陳冠穎

Yokoi connectivity number

[illegible]

說明：

downSampling

```
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

```
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

```
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
```

fFunc 照講義上的 f function 判斷 q 的數量

```
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。

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