R10521516 廖浚評

Use python OpenCV

```
def img_binarize(img_in ):
    shape = img_in.shape
    binimg = np.zeros(shape )
    for i in range(shape[0]):
        for j in range(shape[1]):

                if img_in[i][j] >= 128:
                    binimg[i][j] =1
                else:
                    binimg[i][j] = 0
    return binimg
```

之前做過的二值化

```
def img_downsample(img_in):
    #Downsampling Lena from 512x512 to 64x64,
    img_down = np.zeros((64,64), dtype =int)
#then using 8x8 blocks as a unit, take the topmost-left pixel as the downsampled data.
    for i in range(img_down.shape[0]):
        for j in range(img_down.shape[1]):
            img_down[i][j] = img_in[8*i][8*j]
    return img_down
```

縮小

```
# def yokoi h(bcde) q r s while 4 connected by def
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'
    else:
        return 's'
def f(a1,a2,a3,a4):
    if a1 == 'r' and a2 == 'r' and a3 == 'r' and a4 == 'r':
        ans = 5
    else:
        ans = 0
        for a_i in [a1, a2, a3, a4]:
            if a_i == 'q':
                ans += 1
    return ans
```

Yokoi 的定義 by 上課講義

```
def getNeighbors( img_in, x, y ):
    return [getValue(img_in, x-1, y+1), getValue(img_in, x, y+1), getValue(img_in, x+1, y+1),
            getValue(img_in, x-1,y), getValue(img_in, x,y), getValue(img_in, x+1, y),      \
            getValue(img_in, x-1,y-1), getValue(img_in, x,y-1), getValue(img_in, x+1 ,y-1)]
```

用 getValu()拿 neighbor 的數值 加自己共 9 個

```
def getValue(img_in , x ,y):
    if x >=img_in.shape[0] or x <0 or y >= img_in.shape[1] or y <0:
        return 0
    return img_in[x][y]
```

拿 binary 的值 判斷式判斷邊界的問題

```
def yokoi(img_in):
    row ,col = img_in.shape
    yokoilist= []
    for i in range(row):
        tmpList = []
        for j in range(col):
            if img_in[i][j] >0:
                ## 3x3 neighbors
                    # x7|x2|x6
                    # x3|x0|x1
                    # x8|x4|x5
                [x7,x2,x6,x3,x0,x1,x8,x4,x5] =  getNeighbors(img_in, i, j)

                a1 = h(x0, x1, x6, x2)
                a2 = h(x0, x2, x7, x3)
                a3 = h(x0, x3, x8, x4)
                a4 = h(x0, x4, x5, x1)

                ans = f(a1,a2,a3,a4)
                print ('%d' % ans , end = '')
                tmpList.append(ans)
            else:
                print (' ', end='')
                tmpList.append(0)
        yokoilist.append(tmpList)
    return yokoilist
```

讀入圖片一個一個去抓 neighbor 去判斷

tmpList 是一個一個 row 去存 之後再整 row 存進去 yokoilist 裡面

```
img = cv2.imread('lena.bmp', 2)
binimg = img_binarize(img)
imgdown = img_downsample(binimg)
yokoilist = yokoi(imgdown)
file = open("Yokoi.txt", "w")
for i in range(64):
    for j in range(64):
        if yokoilist[i][j] ==0:
            file.write(' ')
        else:
            file.write(str(yokoilist[i][j]))

    file.write('\n')
file.close()
```

最後就是跑一遍再存入 txt 裡面

結果

yokoi.txt - 記事本

檔案(F)　編輯(E)　格式(O)　檢視(V)　說明