Computer Vision_HW6

R09922093 楊子萱

In this homework, I use python3 and import cv2 to read and write images and use cv2 function to translate image into the binary image. To run my code, use command line and enter python3 hw6.py (http://hw6.py) .

(a) Downsampling Lena
First resize the lena.bmp from 512 * 512 to 64 *64 and change it to binary image. than save it as down_sample.bmp

```
lena = cv2.imread('lena.bmp', 0)
down_row = lena.shape[0]//8
down_column = lena.shape[1]//8
print(down_row, down_column)
down_sampling = np.full((down_row,down_column),0)
for i in range (down_row):
    for j in range (down_column):
        down_sampling[i][j] = lena[8 * i][8 * j]
        if down_sampling[i][j] < 128:
            down_sampling[i][j] = 0
        else :
            down_sampling[i][j] = 255

cv2.imwrite("down_sample.bmp",down_sampling)
```

DownSampling Result:

(b) Yokoi connectivity number

1. Follow the rule in the lecture ppt, that is

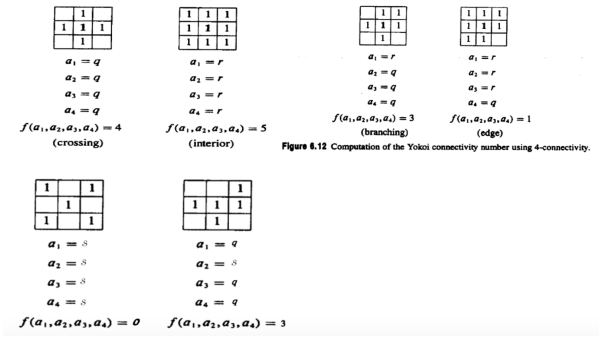

Figure 8.12 Computation of the Yokoi connectivity number using 4-connectivity.

```
def getvalue(i,j,check):
    if check == 1:
        if j+1 >= down_column:
            return 0
        else:
            return down_sampling[i][j+1]
    elif check == 2:
        if i-1 < 0:
            return 0
        else:
            return down_sampling[i-1][j]
    elif check == 3:
        if j-1 < 0:
            return 0
        else:
            return down_sampling[i][j-1]
    elif check == 4:
        if i+1 >= down_row:
            return 0
        else:
            return down_sampling[i+1][j]
    elif check == 5:
        if i+1 >= down_row or j+1 >= down_column:
            return 0
        else:
            return down_sampling[i+1][j+1]
    elif check == 6:
        if i-1 < 0 or j+1 >= down_column:
            return 0
        else:
            return down_sampling[i-1][j+1]
    elif check == 7:
        if i-1 < 0 or j-1 < 0:
            return 0
        else:
            return down_sampling[i-1][j-1]
    elif check == 8:
        if i+1 >= down_row or j-1 < 0:
            return 0
        else:
            return down_sampling[i+1][j-1]
```

2. Follow the formula mentioned in the lecture note.
   Formular1:

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

```
def h(i,j,p1,p2,p3):
    v_0 = down_sampling[i][j]
    v_1 = getvalue(i,j,p1)
    v_2 = getvalue(i,j,p2)
    v_3 = getvalue(i,j,p3)
    if v_0 != v_1:
        return 2
    elif v_0 == v_2 and v_0 == v_3:
        return 1
    else:
        return 0
```

Formula2:

$$f(a_1,a_2,a_3,a_4) = \begin{cases} 5 & \text{if } a_1 = a_2 = a_3 = a_4 = r \\ n & \text{where } n = \#\{a_k | a_k = q\}, \text{ otherwise} \end{cases}$$

```
def f(a,b,c,d):
    if a==b and b==c and c==d and d == 1:
        return 5
    cnt = 0
    if a == 0:
        cnt+=1
    if b == 0:
        cnt+=1
    if c == 0:
        cnt+=1
    if d == 0:
        cnt+=1
    return cnt
```

(c) I use cv2 function write the result number into a image

```
def addnumber_to_photo(image, row, column, number):
        txt = str(number)
        x = column * 15
        y = (row +1 ) * 15
        cv2.putText(image, txt, (x,y), cv2.FONT_HERSHEY_COMPLEX_SMALL,
    1, 0, 1, cv2.LINE_AA)
```

Result: