Computer Vision hw6

B05902050

黃子源

I use PIL to complete the homework. In my program, I use function getpixel() and putpixel() to get the value of every pixel.
For the 4-connectivity neighborhood, I create a list which represent all 8 points around center (0, 0).

Binarize the benchmark and take the topmost-left pixel as the down-sampled data:
In this part, I use the same method as I used in homework 2, And create a 64*64 image which called img3 and put all the topmost-left pixels of 8*8 block in it.

Principal code fragment:

```
img=(Image.open("lena.bmp")).convert("L")
img2 = Image.new("1", (512, 512))
neighbor = [(1,0),(0,-1),(-1,0),(0,1),(1,1),(1,-1),(-1,-1),(-1,1)]
for i in range(512):
    for j in range(512):
        if img.getpixel((i,j)) < 128:
            img2.putpixel((i, j), 0)
        elif img.getpixel((i,j)) >= 128:
            img2.putpixel((i, j), 1)
img3 = Image.new("1", (64, 64))
for i in range(64):
    for j in range(64):
        img3.putpixel((i, j), img2.getpixel((i*8, j*8)))
```

Count the Yokoi connectivity number and store the data as 64*64 matrix:
For all the ‘1’ value pixel in img3, record all its 8 neighbor pixel values as a list named label. After that, I create a function that can return the current number of q and r after the h(b, c, d, e) calculation, so just simply call the function four times and we can get the number of q and r after knowing a1 to a4.
If we have four r, then write ‘5’ in a txt file; otherwise, if the number of q is not 0, then write the number of q in the file; else, write a ‘ ‘ blank character in the file.

Principal code fragment:

```
from PIL import Image, ImageFile, ImageDraw

def h(b,c,d,e,countq,countr):
    if b == c and b == d and b == e:
        return (countq, countr + 1)
    elif b != c:
        return (countq, countr)
    else :
        return (countq + 1, countr)
```

```
f = open("yokoi.txt", "w")

for j in range(64):
    for i in range(64):
        if img3.getpixel((i,j)) == 1:
            label = [0 for i in range(8)]
            countq = 0
            countr = 0
            for k in range(8):
                x = i + neighbor[k][0]
                y = j + neighbor[k][1]
                if 0 <= x < 64 and 0 <= y < 64:
                    label[k] = img3.getpixel((x,y))
                else:
                    label[k] = 0
            (countq, countr) = h(1, label[0], label[1], label[5], countq, countr)
            (countq, countr) = h(1, label[1], label[2], label[6], countq, countr)
            (countq, countr) = h(1, label[2], label[3], label[7], countq, countr)
            (countq, countr) = h(1, label[3], label[0], label[4], countq, countr)
            if countr == 4:
                f.write('5')
            elif countq != 0:
                f.write(str(countq))
            else :
                f.write(' ')
        else:
            f.write(' ')
    f.write('\n')
```

Output of the Yokoi connectivity number:

```
11111111         12111111111122322221     111111111111
15555551          115555555511 2 11  11   1155555555511
15555551         1 2115555112  21112221    155555555551       21
15555551         1 2 155112 22221511        1555555555511      1
15555551          22 2112 22    121         15555555555511
15555551          1  2  21 2     1   1      15555555555551
15555551            12 1  121111    1321   155555555555511
15111551            1322 1155551111        155555555555551
111 1551             1  121555555511       155555555555511
11  1551                 21155555511       15511155555511
21  1551                 2 15555555111     1551 11555511
1   1551                 2 155555555511    1551  115551          1
    1551                1121155555555551   1551   15511         12
    1551                15555555555555511  1551   1111         111
    1551         1     2221155555555555511 1151    11         1151
    1551         2    22 1 1555555555555511 151  11111       1551
    1551         2    1   11555555555555551 151 115551      11551
    1551         2       1155555555555555511511155511     115551
    1551        12       1155555555555555555555555551      155551
    1551        11      2215555555555555555555555555112     1155551
    1551        111    22 15555555555555555555555551 1     1555551
    1551        1511   1 125112111112111555555555111      11555551
    1551        15521   1 121 1 11  1  15555555111        15555551
    1551        1151  132 2           1155555111         115555551
    1551         151    322          115555111  121      155555551
    1551         1221   2            1555551   131      1155555551
    1551          2     1           115555511   1       1155555551
    1551          2                1155555551          1 155555551
    1551          2               11555555551          21155555551
    1551          1              115555555551          15555555551
    1551           1            11511115555521  1     115555555551
    1551         1 1           11111  1155511   2     155555555551
    1551        131            111     15111    2     155555555551
    1551       121           1121   1  111  1   2    1155555555551
    1551       11            111 1  221 11  1   2    1555555555551
    1551     12        1     21 121  11 1111    2    1555555555551
    1551      1       12    22  151111111551    2   11555555555551
    1551   1                2   1555551115511   1   15555555555551
    1551   2               22  12555551 15551    1  15555555555551
    1551   1               1    1555511 11511    2 115555555555551
    1551                 21     155551 1 151     2 155555555555551
    1551                 2      15555112 151     2 155555555555551
    1551          1   1 1     1155555511111     2 155555555555551
    1551          2  22       111511111212      21155555555555551
    1551          1 12          151    2 1      15555555111555551
    1551                        1111  121       155555551 1555551
    1551                         11111111       155555551 1555551
    1551                           115551       155555551 1555511
    1551                            15551       211111111 155511
    11521       1   12           122155511       2     11 115511
1    151        1    1              155555111     2111     15511
22   1511            1              15555555111   155111   1511
 22  1511            1              15555555551   155551  1151
  2  151                1          11155555555511  155511  1511
  2  1521               1          155555555555511 15551 12151
  2  151              121          155555555555551 155511 1551
  2  1511                          155555555555551 115551 1511
  21 1511               11         155555555555551  111111151
  11 151                          11555555555555511    111511
  11 151                          15555555555555551      151
  11 151                         115555555555555551      211
  11 151                         1155555555555555511     1
  11 151                          155555555555555551
  11 111                         12111111111111111111
```