Homework 6

Yokoi Connectivity Number

R09922063 鄭筠庭 資工所碩一

Programming language: Python 3.7.3

Library used for this homework:

- ✦ Numpy: for array usage
- ✦ Opencv: to read and write the image file

Image info: lena.bmp [512(width),512(height),1(Grayscale)]

Code explanation:

First, binarize the benchmark image lena as in HW2.

```
def binary(image):
    for r in range(rows):
        for c in range(columns):
            if image[r,c] < 128:
                image[r,c] = 0
            else:
                image[r,c] = 255
    return image
```

Second step is to down sampling the binary image from a 512x512 to 64x64. The row and column value are divided by 8, so the parameter “divide” is set to 8. Take the topmost-left pixel in each 8x8 region as the down sampled data.

```
def downsampling(image, divide, smallImage):
    for r in range(0, rows, divide):
        for c in range(0, columns, divide):
            smallImage[int(r/divide), int(c/divide)] = image[r,c]
```

Call the Yokoi function to classify the way a pixel is connected to its like neighbors. Go through every pixel and check its value. If the pixel is white (255), keep go down the function.

```
def Yokoi(image):
    resultImage = np.zeros(image.shape, np.uint8)
    for r in range(rows):
        for c in range(columns):
            if image[r,c] == 0:
                continue

            qNum = 0
            sNum = 0
```

Skip the special cases where the mask is out of image border (<0 or >255).
Check the corner neighborhood with the following conditions, and record how many "q" and "s" this pixel gets.

for 4-connectivity

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

```
for h in range(1,5): # 4-connected
    if copyImage[r+1,c+1] != copyImage[r+1+mask[h][0], c+1+mask[h][1]]:
        sNum += 1
    if (copyImage[r+1,c+1] == copyImage[r+1+mask[h][0], c+1+mask[h][1]]) and (copyImage[r+1,c+1] !=
        copyImage[r+1+mask[h+1][0], c+1+mask[h+1][1]] or copyImage[r+1,c+1] != copyImage[r+1+mask[h+5][0],
        c+1+mask[h+5][1]]):
        qNum += 1
```

Label the image pixels with this formula. Finally, output the number-labeled image to a text file.

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

```
if qNum == 4:
    resultImage[r,c] = 4
elif qNum == 3:
    resultImage[r,c] = 3
elif qNum == 2:
    resultImage[r,c] = 2
elif qNum == 1:
    resultImage[r,c] = 1
elif qNum == 0 and sNum == 0:
    resultImage[r,c] = 5
```

```
f = open("result.txt", 'w')
for r in range(rows):
    for c in range(columns):
        if resultImage[r,c] == 0:
            f.write("  ")
        else:
            f.write(str(resultImage[r,c]))
            f.write(" ")
    f.write('\n')
f.close()
```

Result: