Computer Vision 2019 Fall

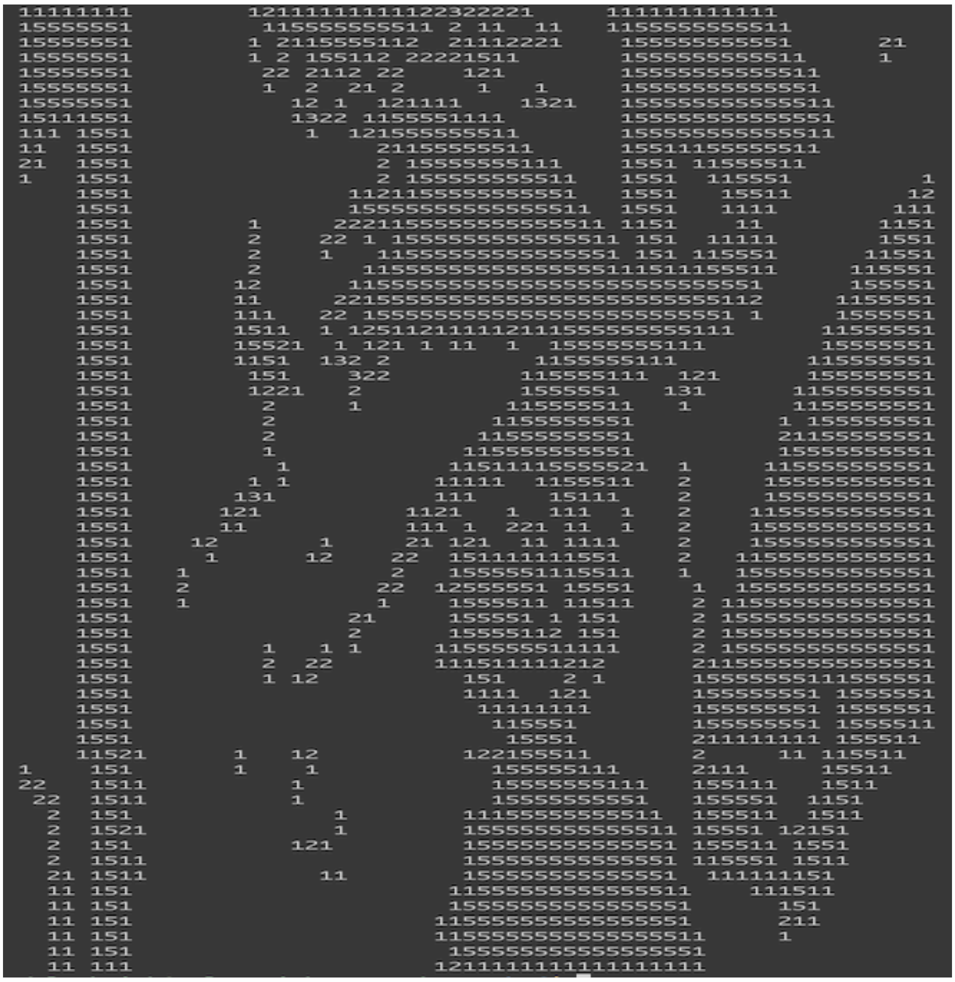
Homework #6

b06902059 資工三 謝宜儒

Description

This homework focuses on calculating the Yokoi connectivity number on a downsampled image.

Results



```
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      15511555555511
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       11555555555555551115111555511     115551
    1551       12      1155555555555555555555555551      155551
    1551       11     221555555555555555555555555112     1155551
    1551       111   22 155555555555555555555555551 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      15555555115555551
    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                     1211111111111111111
```

(Binarize → Downsample → Calculate Yokoi connectivity number)

Source Code

```
"""
Computer Vision 2019 Fall
Homework #6
"""

import sys
import cv2
import numpy as np

def binarize(img):
  binarized_img = np.zeros((length, width))
  threshold = 128
  for i in range(length):
    for j in range(width):
      if(img[i][j] < threshold):
        binarized_img[i][j] = 0
      else:
        binarized_img[i][j] = 255
  return binarized_img

def downsample(img, scale):
  downsampled_img = np.zeros((int(length / scale), int(width / scale)), dtype
= np.uint8)
  for i in range(0, length, scale):
    for j in range(0, width, scale):
      downsampled_img[int(i / scale)][int(j / scale)] = img[i][j]
  return downsampled_img

def h(b, c, d, e):
  if(b == c and (d != b or e != b)):
    return 1
  if(b == c and (d == b and e == b)):
    return 2
  if(b != c):
    return 3

def yokoi_num(img):
  padding_img = np.zeros((img.shape[0] + 2, img.shape[1] + 2), dtype =
np.uint8)
  padding_img[1:-1, 1:-1] = img
  yokoi_graph = np.zeros(img.shape, dtype = np.uint8)

  # -1 for "undefined" (border points), 0 for default
  # 1 for q, 2 for r, 3 for s
  for i in range(1, padding_img.shape[0] - 1):
    for j in range(1, padding_img.shape[1] - 1):
      if(padding_img[i][j] == 0):
        continue
      a1 = h(padding_img[i][j], padding_img[i][j + 1], padding_img[i - 1][j +
1], padding_img[i - 1][j])
```

```
      a2 = h(padding_img[i][j], padding_img[i - 1][j], padding_img[i - 1][j - 1], padding_img[i][j - 1])
      a3 = h(padding_img[i][j], padding_img[i][j - 1], padding_img[i + 1][j - 1], padding_img[i + 1][j])
      a4 = h(padding_img[i][j], padding_img[i + 1][j], padding_img[i + 1][j + 1], padding_img[i][j + 1])
      count_q = (a1 == 1) + (a2 == 1) + (a3 == 1) + (a4 == 1)
      count_r = (a1 == 2) + (a2 == 2) + (a3 == 2) + (a4 == 2)
      if(count_r == 4):
        yokoi_graph[i - 1][j - 1] = 5
      else:
        yokoi_graph[i - 1][j - 1] = count_q

  return yokoi_graph

def print_graph(graph):
  for i in range(graph.shape[0]):
    for j in range(graph.shape[1]):
      if(graph[i][j] != 0):
        print(graph[i][j], end = '')
      else:
        print(' ', end = '')
    print('')

if __name__ == "__main__":
  if(len(sys.argv) != 2):
    printf("Usage: python3 hw6.py [input image]\n")

  # Read the input image in grayscale mode
  img = cv2.imread(sys.argv[1], cv2.IMREAD_GRAYSCALE)
  length, width = img.shape[0], img.shape[1]

  binarized_img = binarize(img)
  downsampled_img = downsample(binarized_img, 8)
  yokoi_graph = yokoi_num(downsampled_img)
  print_graph(yokoi_graph)
```

To run the source code, type the following line in a terminal:

```
python3 hw6.py [input image]
```

where in this homework, the input image is **lena.bmp**.