HW6 R11944009 何明倩

(a) Yokoi connectivity number

- Description:
  針對 lena 轉換成 binary image 並 downsmaple 成 64x64。之後找 Yokoi connectivity number
- Algorithm:
  Step 1: binary image 和 downsample
  將 lena binary image, 做 8x8 的 window sliding，每次移動 8 pixels，取這個 window 中左上角的值當代表。

  Step 2: 建立 h function

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

  可以根據此規則建立 h function

  Step 3: 建立 f function

$$f(a1,a2,a3,a4) = \begin{cases} 5 \; if \; a1 = a2 = a3 = a4 = 0 \\ n \; where \; the \; number \; of \; q \end{cases}$$

  可以根據此規則建立 function。

  Step 4: 計算 Yokoi connectivity number
  先針對 step1 的影像進行 zero padding 1 pixel 成為 new image，向左向下拜訪 new image 的每個 element，當該 pixel 值為 1，則根據以下方程式計算出 a1, a2, a3, a4。

$$a_1 = h(x_0, x_1, x_6, x_2)$$
$$a_2 = h(x_0, x_2, x_7, x_3)$$
$$a_3 = h(x_0, x_3, x_8, x_4)$$
$$a_4 = h(x_0, x_4, x_5, x_1)$$

  在計算$f(a1,a2,a3,a4)$即可以得到該 pixel 的 Yokoi connectivity number。

- Principal code fragment:
  - down sample

```
def downsample(img):
    scale = 8
    h, w = img.shape
    img_out = np.zeros((h//scale, w//scale))

    for i in range(h//scale):
        for j in range(w//scale):
            img_out[i, j] = img[i*scale, j*scale]

    return img_out
```

  - h function

```
def h_f(b, c, d, e):
    out = ''
    if ((b == c) and ((d != b) or (e != b))):
        out = 'q'
    elif ((b == c) and ((d == b) and (e == b))):
        out = 'r'
    else:
        out = 's'
    return out
```

  - f function

```
def f(a1, a2, a3, a4):
    data = np.array([a1, a2, a3, a4])
    r_num = np.sum(data == 'r')
    q_num = np.sum(data == 'q')
    out = 0
    if r_num == 4:
        out = 5
    else:
        out = q_num

    return out
```

  - Yokoi connectivity number

```
def Yokoi_4connected(img):
    h, w = img.shape
    img_tmp = np.zeros((h+2, w+2))
    img_tmp[1:1+h, 1:1+w] = img
    output = np.zeros((h, w))
    for i in range(h):
        for j in range(w):
            x = i + 1
            y = j + 1
            if (img_tmp[x, y] > 0):
                a1 = h_f(img_tmp[x, y], img_tmp[x, y+1], img_tmp[x-1, y+1], img_tmp[x-1, y])
                a2 = h_f(img_tmp[x, y], img_tmp[x-1, y], img_tmp[x-1, y-1], img_tmp[x, y-1])
                a3 = h_f(img_tmp[x, y], img_tmp[x, y-1], img_tmp[x+1, y-1], img_tmp[x+1, y])
                a4 = h_f(img_tmp[x, y], img_tmp[x+1, y], img_tmp[x+1, y+1], img_tmp[x, y+1])
                output[i, j] = f(a1, a2, a3, a4)

    return output
```

- Result: