R09922190 王祥任

CV Homework 6

Abstract

本次作業中，我們要先將二值化過的圖片取樣為　64x64　大小，並計算白色像素的　Yokoi connectivity number (4-connectivity)。以上操作除輸入、輸出以外，不得直接套用現成套件。

Implementation

- Programming Language: Python3
- Python Package: opencv-python, matplotlib, numpy, copy
- Execution: python3 hw6-main.py

在執行時，請務必將 lena.bmp 放置在和 hw6-main.py 相同的檔案目錄下，並確認檔名相同。opencv-python用於讀取和寫出圖片，matplotlib僅用於runtime時即時顯示和儲存圖片，須配合jupyter使用，在此不贅述，而最後的copy和numpy則用於純計算上。

Downsample

```python
def downsample(bina):
    after = np.zeros((64, 64))
    for r in range(after.shape[0]):
        for c in range(after.shape[1]):
            after[r][c] = bina[r*8][c*8]
    return after
```

根據要求，我們要將二值化後的圖片downsample為原先的1/8，即64x64大小。以8x8的block為一單位，取每個block的最左上角的pixel即可。

Yokoi Connectivity Number

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

- for 4-connectivity

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

```python
def f(a_list):
    r_num = 0
    q_num = 0
    for a in a_list:
        if a == 'q': q_num +=1
        if a == 'r': r_num +=1
    if (r_num == 4): return 5
    else: return q_num
```

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

```python
def yokoi(downa):
    after = np.zeros((64, 64))
    blocks = [
        [(0, 0), ( 0,  1), (-1,  1), (-1,  0)],
        [(0, 0), (-1,  0), (-1, -1), ( 0, -1)],
        [(0, 0), ( 0, -1), ( 1, -1), ( 1,  0)],
        [(0, 0), ( 1,  0), ( 1,  1), ( 0,  1)],
    ]
    for r in range(downa.shape[0]):
        for c in range(downa.shape[1]):
            if downa[r][c] == 255:
                a = []
                for blk in blocks:
                    p = []
                    for ofst in blk:
                        if (r+ofst[0] >= 0 and r+ofst[0] < downa.shape[0]) and (c+ofst[1] >= 0 and c+ofst[1] < downa.shape[1]):
                            p.append(downa[r + ofst[0]][c + ofst[1]])
                        else:
                            p.append(0)
                    a.append(h(p[0], p[1], p[2], p[3]))
                after[r][c] = f(a)
    return after
```

- $a_1 = h(x_0, x_1, x_6, x_2)$

  $a_2 = h(x_0, x_2, x_7, x_3)$

  $a_3 = h(x_0, x_3, x_8, x_4)$

  $a_4 = h(x_0, x_4, x_5, x_1)$

x_7	x_2	x_6
x_3	x_0	x_1
x_8	x_4	x_5

- $output = f(a_1, a_2, a_3, a_4)$

我們要計算的是白色像素的Yokoi Connectivity Number for 4-connectivity。我們遍歷整張已取樣過的圖片，當遇到白色像素的時候，便對它和它附近的像素(若超出邊界，則預設為黑色)做四種不同輸入的h函數，輸入的排序請參照上方的圖片組，h的實作為參照講義的公式，根據符合的條件回傳q、r、s。而當四種組合的h都計算出來，得到a1、a2、a3、a4後，便可以帶入f函數中取得該白色像素對應的Yokoi Connectivity Number for 4-connectivity。f函數的實作也是參照講義的公式，當a1~4全部都是r的時候回傳5，否則就回傳q的數量。依照此邏輯遍歷完整張64x64的圖片即可。成果顯示於下一頁，圖片當中為0者並沒有顯示。

R09922190 王祥任