Computer Vision HW7

R06922075 翁瑋

這次的作業是實作 Thinning Operator，benchmark 是 64*64 downsampling 過後的 lena.bmp，做法跟 hw6 的一樣，這邊就不贅述 downsampling 的過程。

```python
def _int_bor(im) :
    pixels = im.load()
    int_bor_list = []
    # b border
    # i interior
    for j in range(1,im.size[0]-1) :
        tmp = []
        for i in range(1,im.size[1]-1) :
            if pixels[i,j] == 255 :
                this = 'i'
                for m in range(-1,2) :
                    for n in range(-1,2) :
                        if pixels[i+m,j+n] != 255 :
                            this = 'b'
                            break
                tmp.append(this)
            else :
                tmp.append(' ')
        int_bor_list.append(tmp)
    return int_bor_list
```

Step1 : _int_bor()這個函式，對整張圖進行 label，標記出 interior 跟 border 的部分，回傳一個二維陣列。

```python
def _marked(int_bor) :
    marked_list = []
    for i in range(len(int_bor)) :
        tmp = []
        for j in range(len(int_bor[0])) :
            if int_bor[i][j] == 'b' :
                this = 'b'
                for m in range(-1,2) :
                    for n in range(-1,2) :
                        if i+m < len(int_bor) and i+m >= 0 and j+n < len(int_bor[0]) and j+n >= 0 and int_bor[i+m][j+n] == 'i' :
                            this = 'm'
                            break
                tmp.append(this)
            else :
                tmp.append(int_bor[i][j])
        marked_list.append(tmp)
    return marked_list
```

Step2 : _marked()這個函式，對剛標記好的 int_bor_list 進行搜尋，找出 marked 的部分，標記上’m’，也回傳一個二維陣列。

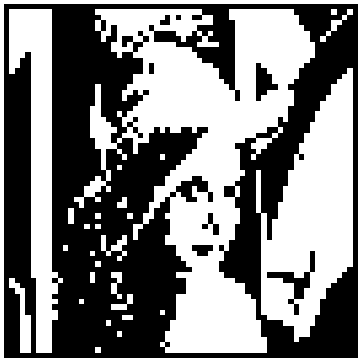
```python
def _shrink(im , marked) :
    im_shrink = im.copy()
    # b border
    # i interior
    for i in range(1,im_shrink.size[0]-1) :
        for j in range(1,im_shrink.size[1]-1) :
            if marked[i-1][j-1] == 'm' :
                if yokoi_pixel(im_shrink,j,i) == 1 :
                    im_shrink.putpixel((j , i) , 0)
    return im_shrink
```

Step3 : _shrink()這個函式，對剛標記好的 marked 跟原圖(64*64)的去做比較，配合 yokoi number 找出可以刪掉的部分，直接對原圖做修改。由左而右，由上到下整個跑完後，回傳修改過的圖。

```
108     while 1 :
109         int_bor = _int_bor(im_down)
110         marked = _marked(int_bor)
111         im_shrink = _shrink(im_down,marked)
112 
113         pixel_before = im_down.load()
114         pixel_after = im_shrink.load()
115         diff = False
116         for i in range(im_down.size[0]) :
117             for j in range(im_down.size[1]) :
118                 if pixel_before[i,j] != pixel_after[i,j] :
119                     diff = True
120                     break
121         if diff == False :
122             break
123         else :
124             im_down = im_shrink
125 
126     im_shrink.save(benchmark+'_shrink.bmp')
```

主程式裡面主要在跑的部分，113~124 行式檢查 shrink 過後的圖跟前一張有沒有不同的部分，若是沒有，則代表 shrink 完畢可以結束工作，否則不斷重複，step1~3。

Result :

原圖	DownSampling	結果
		

Environment :

Anaconda3

Python 3.6.1

Using Library :

PIL

Benchmark :

Lena.bmp