


# 電腦視覺 作業六

指導老師 傅楸善

學生 蔡宇晴

學號 R08945050

	原圖	Dilation
圖 片		結果呈現在末頁
程 式 碼	<pre> def sampling_and_binarize_():     img = cv2.imread("./lena.bmp",0)     h,w = img.shape     template = np.zeros((64,64))     for height in range(int(h/8)-1):         for weight in range(int(w/8)-1):             # print(img[height*8,weight*8])             template[height,weight] = img[height*8,weight*8] </pre>	

```

def count_number(img):
    mask1 = np.array([[1,1],[0,0]])
    mask2 = np.array([[1, 0], [1, 0]])
    mask3 = np.array([[0,0], [1, 1]])
    mask4 = np.array([[0, 1], [0, 1]])
    q = 0
    r = 0
    # 右
    if img[1,2] != 0:
        slice1 = img[0:2,1:3]
        num = np.sum(np.bitwise_and(slice1,mask1))
        if num != 2:q+=1
        else:r+=1
    # 上
    if img[0,1] != 0:
        slice2 = img[0:2,0:2]
        num = np.sum(np.bitwise_and(slice2,mask2))
        if num != 2:q+=1
        else:r+=1
    if img[1,0] != 0:
        slice3 = img[1:3,0:2]
        num = np.sum(np.bitwise_and(slice3,mask3))
        if num != 2:q+=1
        else:r+=1
    # 下
    if img[2,1] != 0:
        slice4 = img[1:3,1:3]
        num = np.sum(np.bitwise_and(slice4,mask4))
        if num != 2:q+=1
        else:r+=1
    if r == 4:return 5
    else:return q

```

```
def yokoi(img):
    # 回傳一個矩陣
    h,w = img.shape
    is_full = False
    template = np.zeros((64,64),dtype = int)
    img = np.pad(img,(1,1))
    for height in range(1,h+1):
        for weight in range(1,w+1):
            if img[height,weight] != 0:
                num = count_number(img[height-1:height+2,weight-1:weight+2])
                template[height,weight] = num
    np.set_printoptions(linewidth=200,threshold = 100000)
    template = np.reshape(template,(64,64))
    print(template)
```

Step1.將圖片 downsampling 再轉成二值化

Step2.對圖片做遍歷，若(l,j)有值，則確認他周圍的 Yokoi 數量

Step3.填入 yokoi 數至 template,print 出並截圖。

**描  
述**

最後結果依然有 lena 的輪廓出現，而且可以明顯看出在平坦區域（如肩膀），數值大多是 5,因為那裡屬於內側。

而邊緣處大多是 1。

而其他數值（q = 2,3,4）會出現在多種物件的交接處，去除後會使物件分開。

[illegible]



