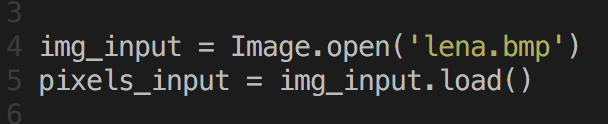
**CV hw2 / 電機所R06921082 陳與賢**

**Description:**

利用python來處理bmp檔，產生binary image / histogram / connected component，使用numpy及PIL這兩個套件。

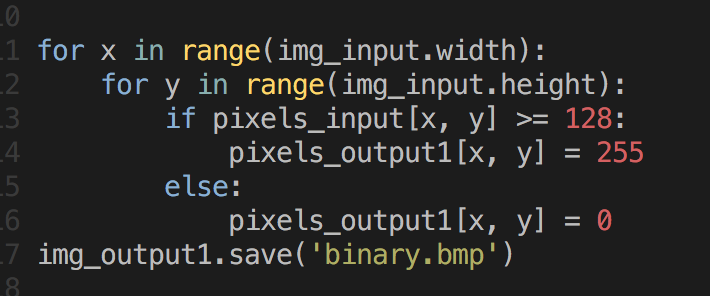
首先讀入圖檔，code如下：



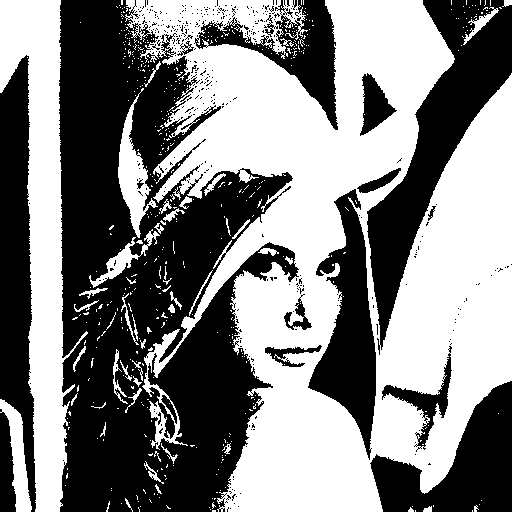
**（一）Binary image**

* *Algorithm*

依據pixels\_input的灰階值，若 ≥ threshold(128)則給值255，反之給0，code如下：



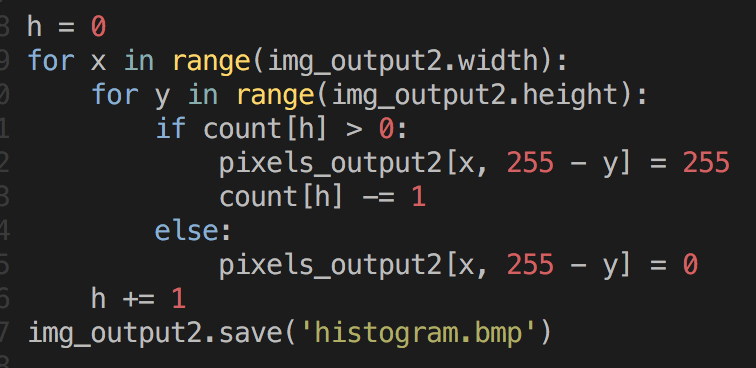
* *Result*



**（二）Histogram**

* *Algorithm*

先開一個大小為256的array，然後遍歷所有pixels來計算0~255值的出現次數，再來找出出現的最多次數，因為output圖大小為256\*256，因此等比例縮小各個值的出現次數，接下來依據下面的code填0或255即可畫出histogram



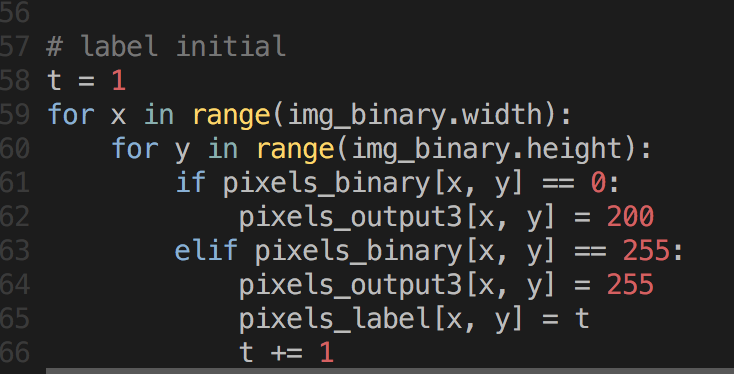
* *Result*



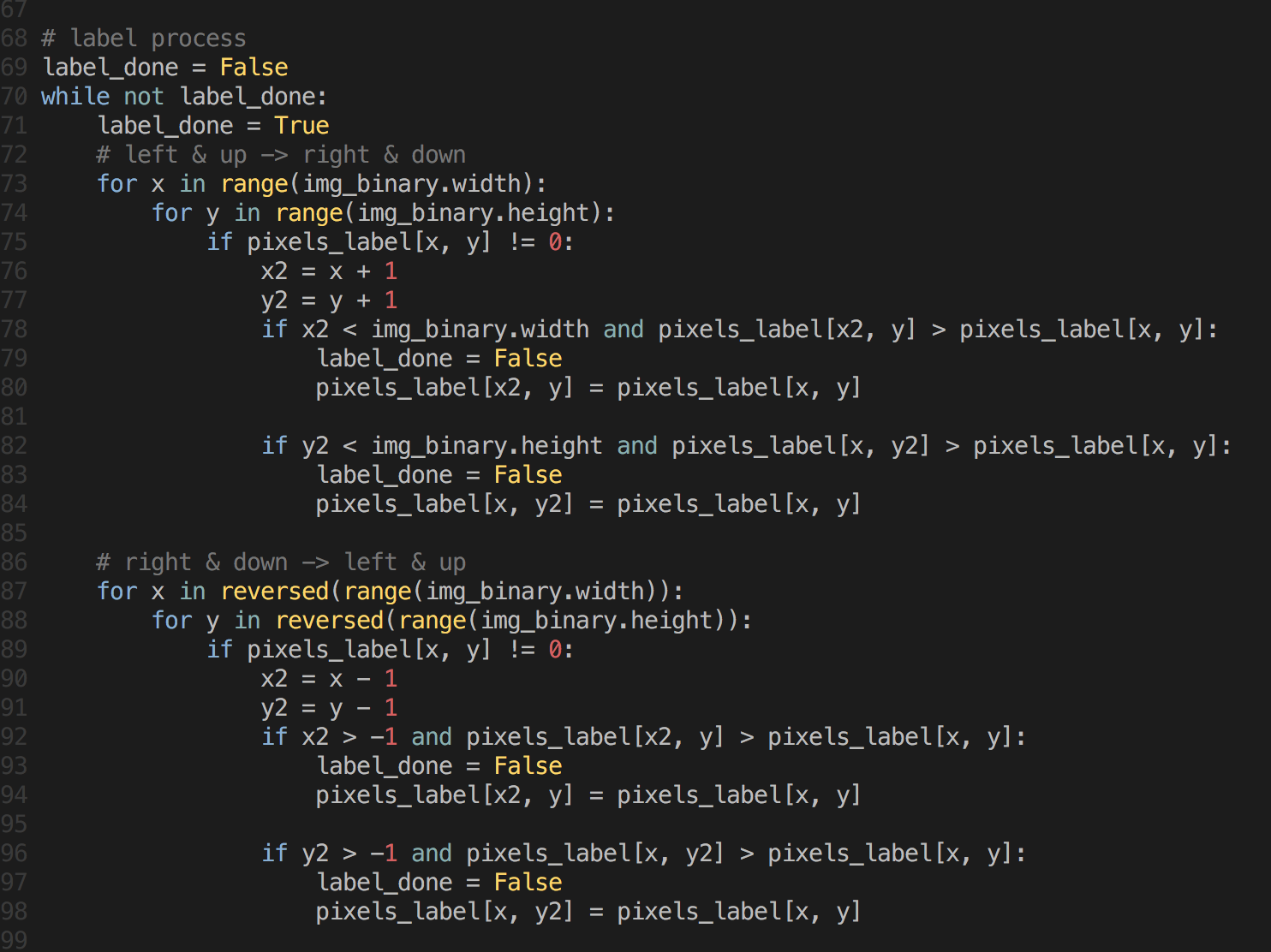
**（三）Connected component**

* *Algorithm*

首先創造一個一樣的圖，只是深色的值從0改成200（因為方便黑色框框較為看得清楚），並且值為255的pixels分別建立unique label值，code如下：



再來開始找出connected component，這邊是使用iterative的方法，用直行來看，從左上到右下再從右下到左上，不斷更新直到沒有更改label值，code如下：



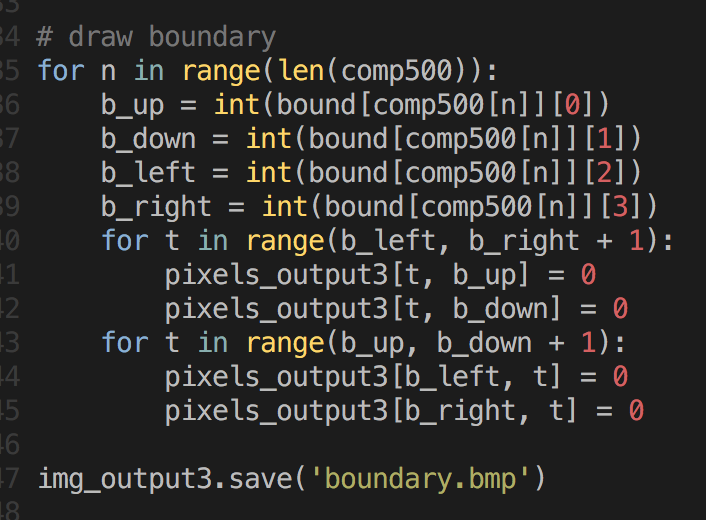
再來找出≥500個pixels的component的label值，並且存進comp500這個list，再來開始找出邊界

主要的想法是每個邊界的值都設成最邊邊，然後依序比較當前的pixel的x,y值，若發現應該更新則更新

例如說在x = 20, y =20時發現label值為1，先檢查1是否屬於≥500個pixels的component，若是，則檢查label = 1的boundary，再來假如發現上界是30，而 x = 20 < 30 所以應該上拉（更新）上界，其餘邊界以此類推，最後即可找出各個component的邊界，code如下：



最後依據此值畫出邊界（給pixel值0 -> 黑色）即可，code如下：



* *Result*

