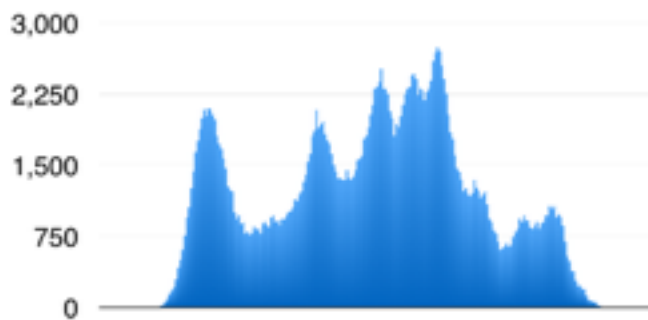

Computer Vision HW2

National Taiwan University

r02944002 王瀚宇



參數：

```
int headerLength = 172;  
int imageWidth = 512;  
int imageHeight = 512;  
int unit = 1;  
int threshold = 128;  
[相鄰判定：4-connected]
```

(a)a binary image (threshold at 128)

基本想法：

用基本IO讀入後，使用For迴圈，把byte數值小於128的都改成0，大於等於128的都改成255。

實作：

```
ArrayList<Integer> bytes = GetByteData(fileName);
for(int y = 0 ; y<imageHeight; y++)
{
    for(int x = 0; x <imageWidth;x++)
    {
        int data = bytes.get(headerLength+(y*imageWidth+x)*unit);

        if (data < threshold) data = 0;
        else data = 255;

        bytes.set(headerLength+(y*imageWidth+x)*unit, data);
    }
}
WriteOut(bytes,"./assets/hw2-a.im");
```

結果：



(b)a histogram

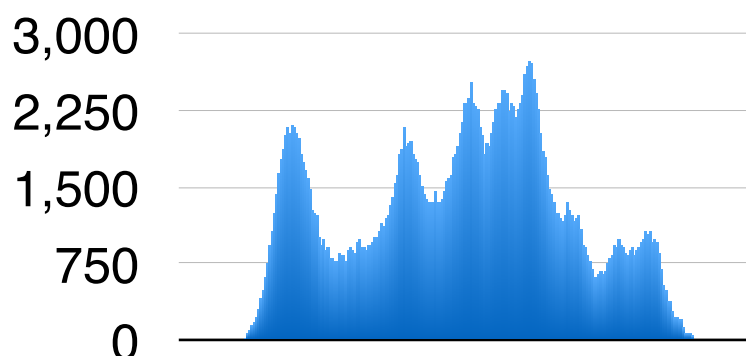
基本想法：

使用JAVA中的HashTable，對0~255建立統計表，然後用for迴圈將每一格的數值統計到對應的位置內。最後將HashTable的Data輸出到CSV檔，使用Numbers來畫圖。

實作：

```
bytes = GetByteData(fileName);
Hashtable<Integer,Integer> hashtable = new Hashtable<Integer,Integer>();
for(int i =0;i<256;i++){hashtable.put(i, 0); }
for(int y = 0 ; y<imageHeight; y++)
{
    for(int x = 0; x <imageWidth;x++)
    {
        int data = bytes.get(headerLength+(y*imageWidth+x)*unit);
        int oldValue = hashtable.get(data);
        int newValue = oldValue + 1;
        hashtable.replace(data, newValue);
    }
}
File f = new File("./assets/hw2-b.csv");
if(f.exists())f.delete();
FileWriter writer = new FileWriter(f);
writer.write("value, count\n");
for(int i = 0 ;i<256;i++)
{
    int value = i;
    int data = hashtable.get(i);
    writer.write(value+","+data+"\n");
}
writer.close();
```

結果：



(c)connected components

基本想法：**[相鄰判定：4-connected]**

幫每一個Pixel編編號，當相鄰的點都是白色時，會統一對方的編號。像是1號跟2號相鄰，又都是白色的話，會變成兩個pixel都標一號。然後用雙重For迴圈檢查所有的Pixel，都跟相鄰的Pixel統一編號。（相鄰不止隔壁，只要白色有連到都變），實作時維持好幾個List，把編號一樣的Pixel放在同樣的List。當兩坨碰在一起時，兩個List也會結合。最後再把List長度超過500的圖片

實作：

```
for(int y = 0 ; y<imageHeight; y++)
{
    for(int x = 0; x <imageWidth;x++)
    {
        connectMapCheck(bytes,records,GroupMap,x,y);
    }
}
```

```
ArrayList<ArrayList<Integer>> candidates = new
ArrayList<ArrayList<Integer>>();
```

```
for(int keyValue : GroupMap.keySet())
{
    ArrayList<Integer> alloc = GroupMap.get(keyValue);

    if(alloc!=null&&alloc.size()>=500)
    {
        candidates.add(GroupMap.get(keyValue));
        System.out.println("keyValue("+keyValue
+"):"+GroupMap.get(keyValue).size());
    }
}
```

```
ArrayList<Rectangle> rectangles = new ArrayList<Rectangle>();
```

```
for(ArrayList<Integer> candidate : candidates)
```

```

    {
        rectangles.add(getRectangle(candidate));
    }

    for(Rectangle rec :rectangles)
    {
        ReverseColor(bytes,rec);
        System.out.println("x1:"+rec.x1+" x2:"+rec.x2+" y1:"+rec.y1+"
y2:"+rec.y2);
    }

    WriteOut(bytes,"./assets/hw2-c.im");
    System.out.println("done");

```

結果：

Bounding Boxes:

x1:118 x2:157 y1:94 y2:237

x1:89 x2:139 y1:237 y2:287

x1:0 x2:31 y1:399 y2:511

x1:0 x2:87 y1:0 y2:511

x1:127 x2:511 y1:0 y2:511

