# 第19章
# 直方圖均衡化 - 增強影像對比度

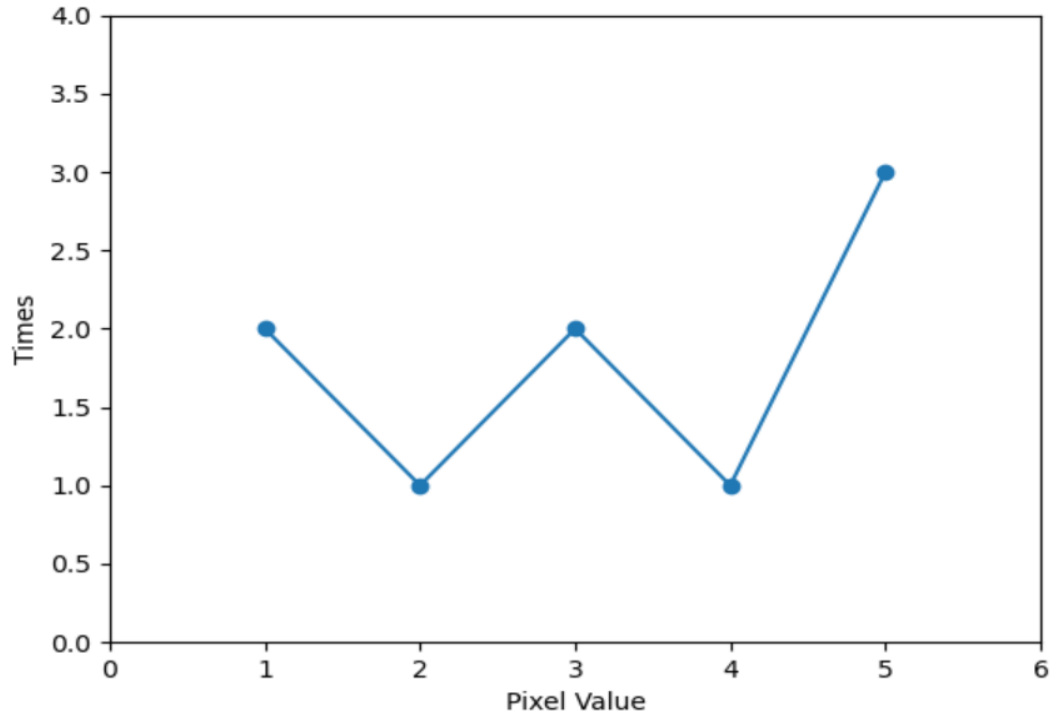
# 19-1：認識直方圖

- 19-1-1：認識直方圖

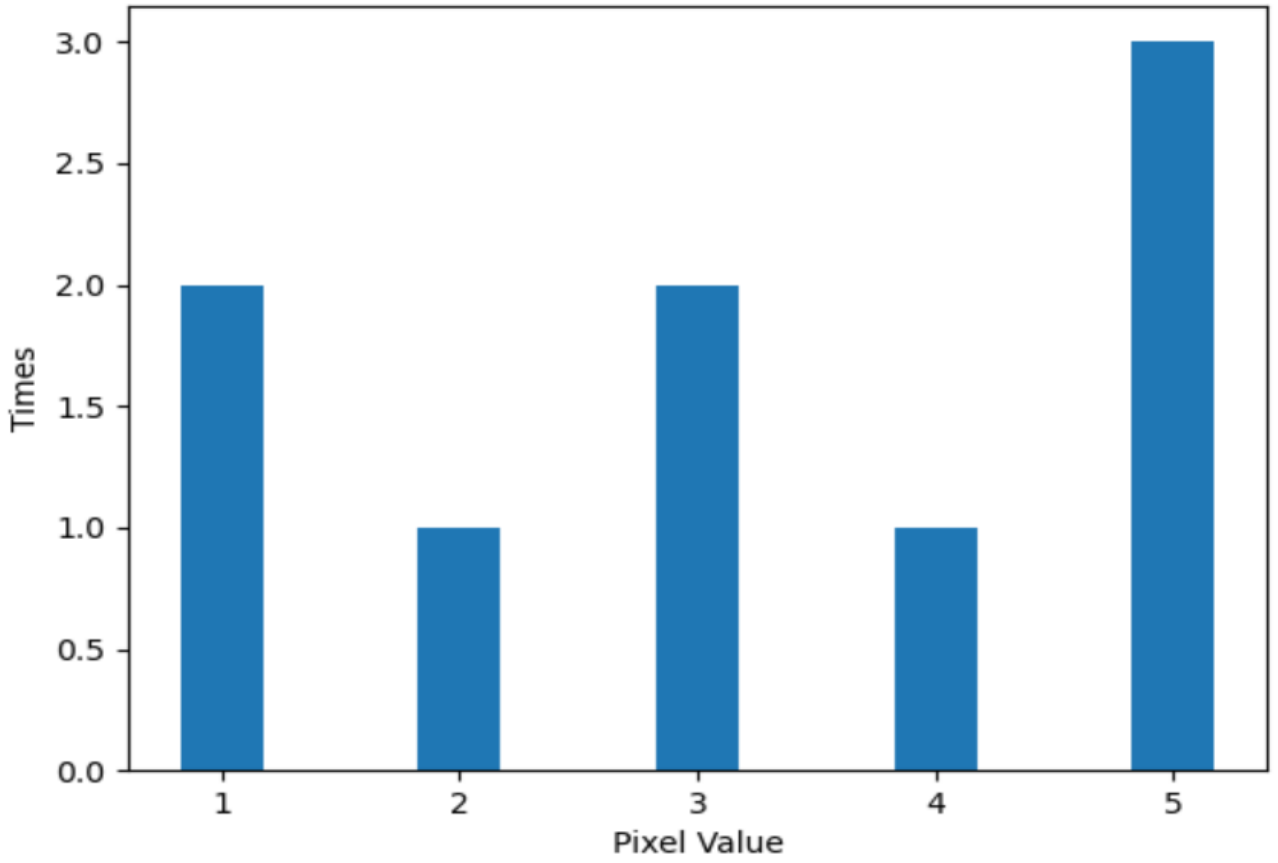
| | | |
|---|---|---|
| 1 | 5 | 1 |
| 3 | 4 | 5 |
| 2 | 3 | 5 |

| 像素值 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 出現次數 | 2 | 1 | 2 | 1 | 3 |

- 程式實例ch19_1.py：使用折線圖plot( )函數，繪製上述像素值出現的次數。



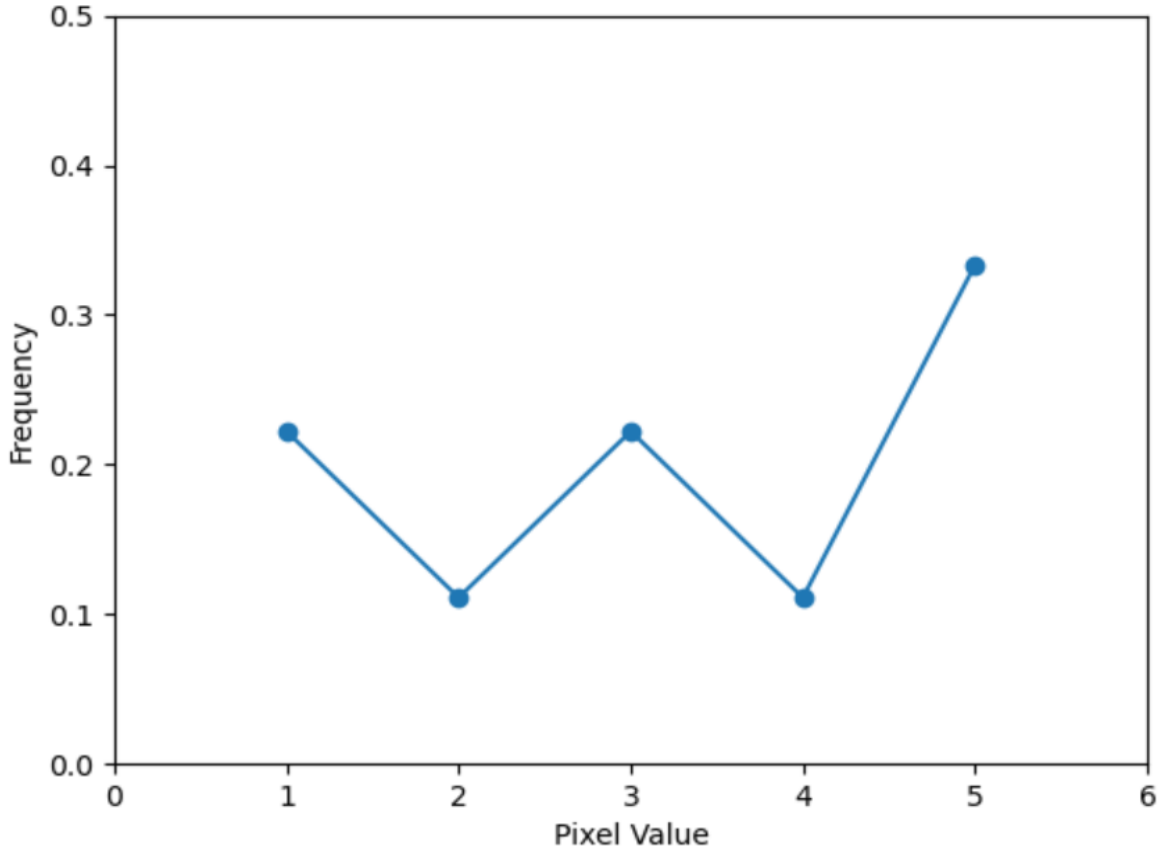
- 程式實例ch19_2.py：使用直方圖bar( )函數重新設計上一個程式，產生長條圖。



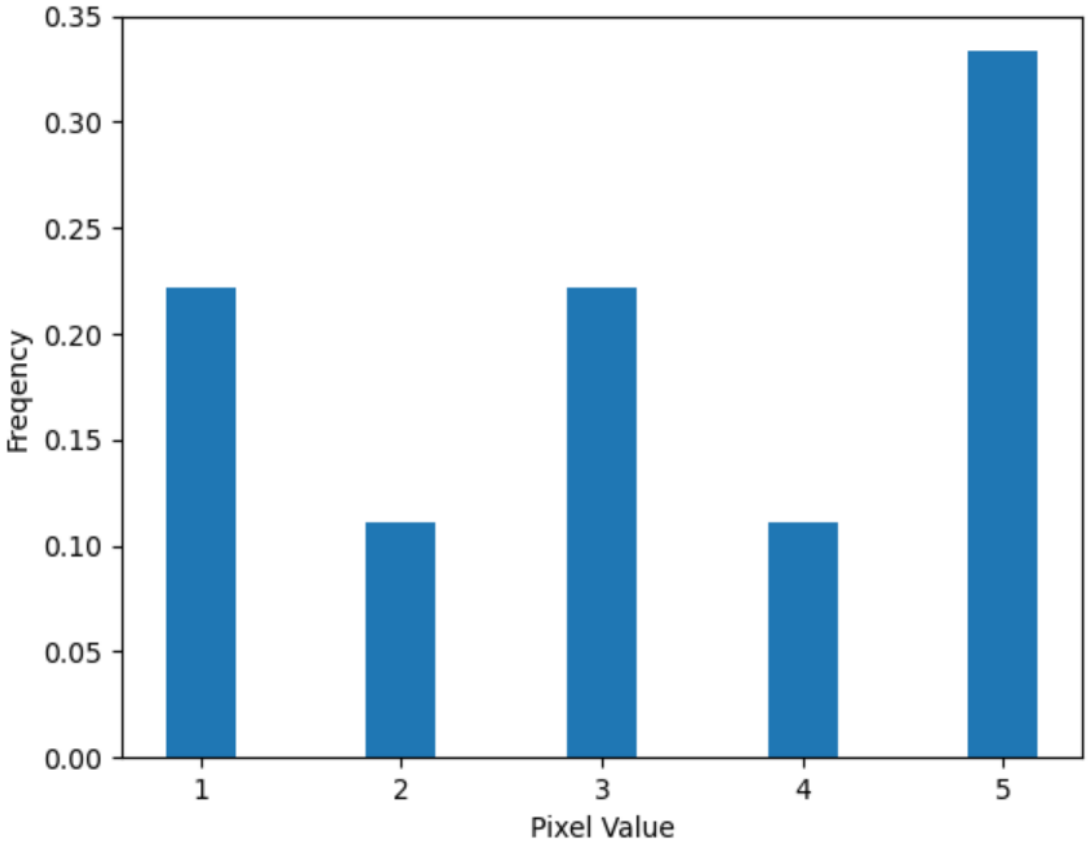
# 19-1-2：歸一化直方圖

| 像素值 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 出現次數 | 2/9 | 1/9 | 2/9 | 1/9 | 3/9 |

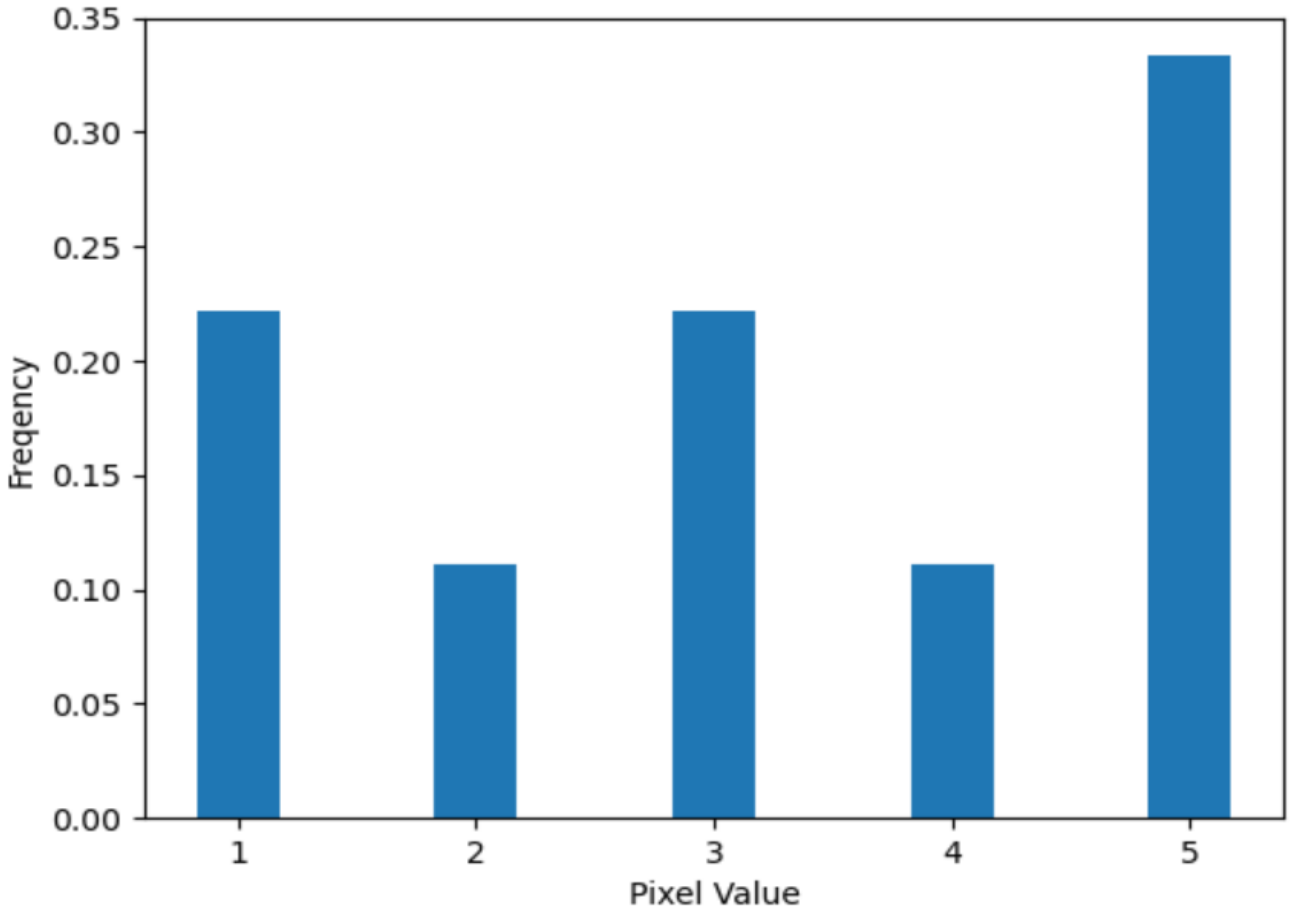
- 程式實例ch19_3.py：使用歸一化觀念重新設計ch19_1.py。


0.5
0.4
0.3
0.2
0.1
0.0
Frequency
0
1
2
3
4
5
6
Pixel Value

- 程式實例ch19_4.py：使用歸一化觀念重新設計ch19_2.py。

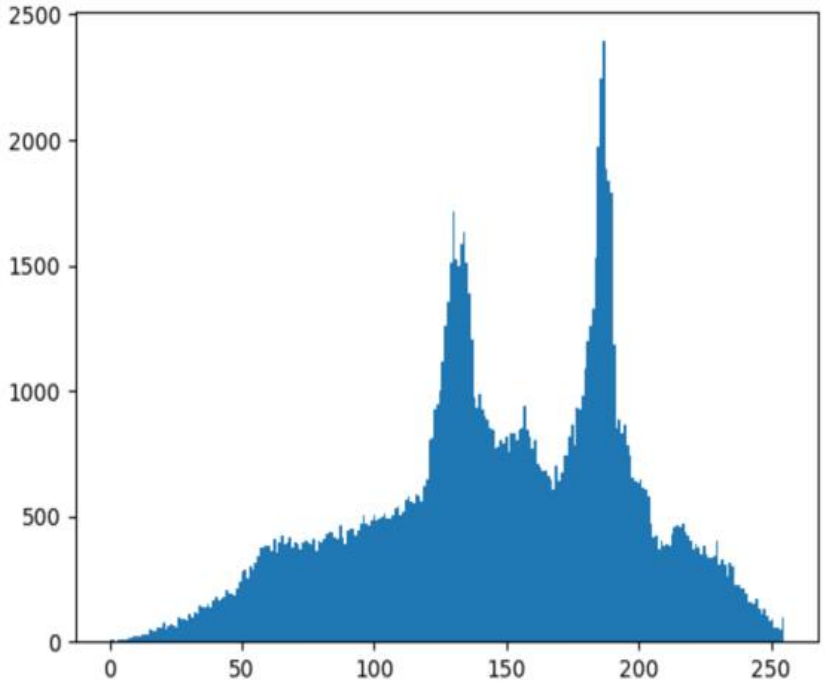


- 程式實例ch19_4.py：使用歸一化觀念重新設計ch19_2.py。

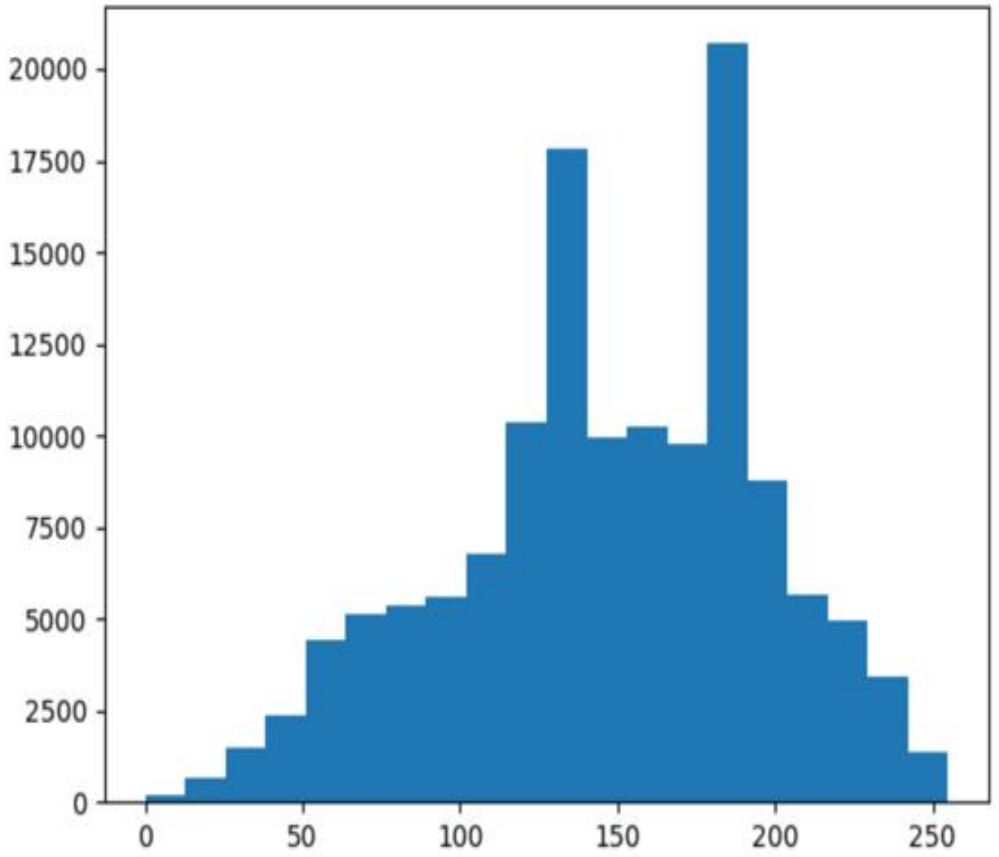


# 19-2：繪製直方圖

- **19-2-1：使用matplotlib繪製直方圖**
- 程式實例ch19_5.py：繪製snow.jpg影像檔案的直方圖。

- 程式實例ch19_6.py：重新設計ch19_5.py，設定20個區間。

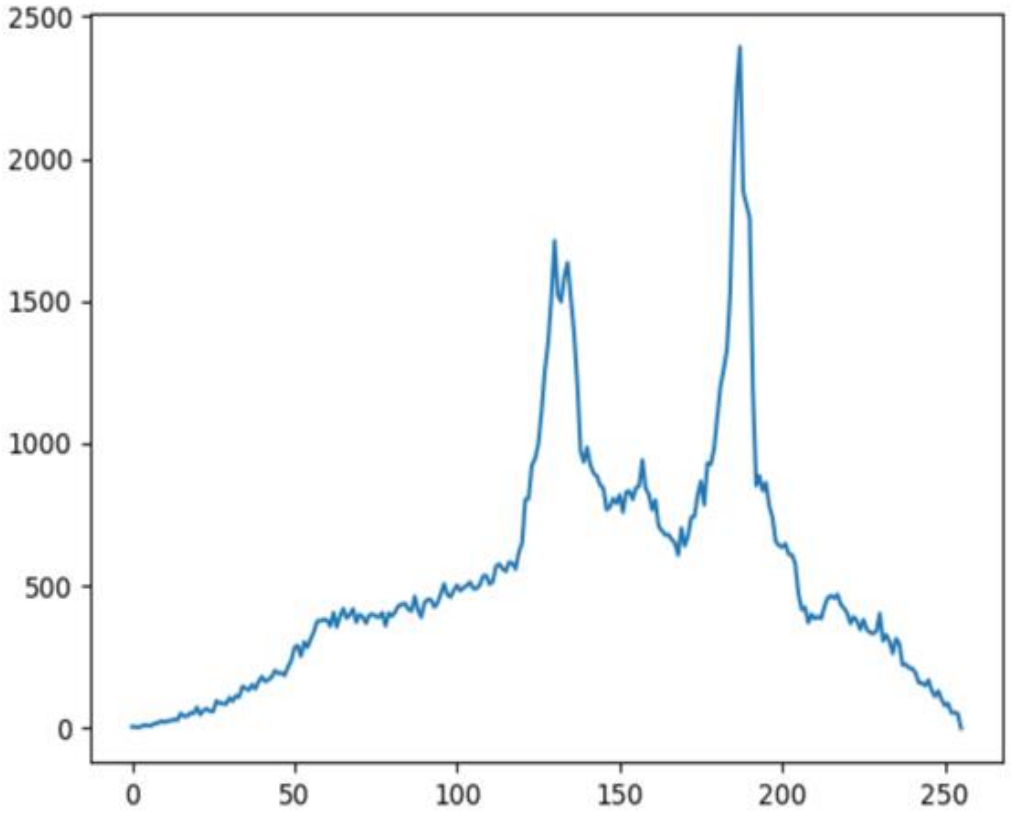



# 19-2-2：使用OpenCV取得直方圖數據

- hist = cv2.calcHist(src, channels, mask, histSize, ranges, accumulate)
- 程式實例ch19_7.py：取得直方圖snow.jpg影像的直方圖數據。

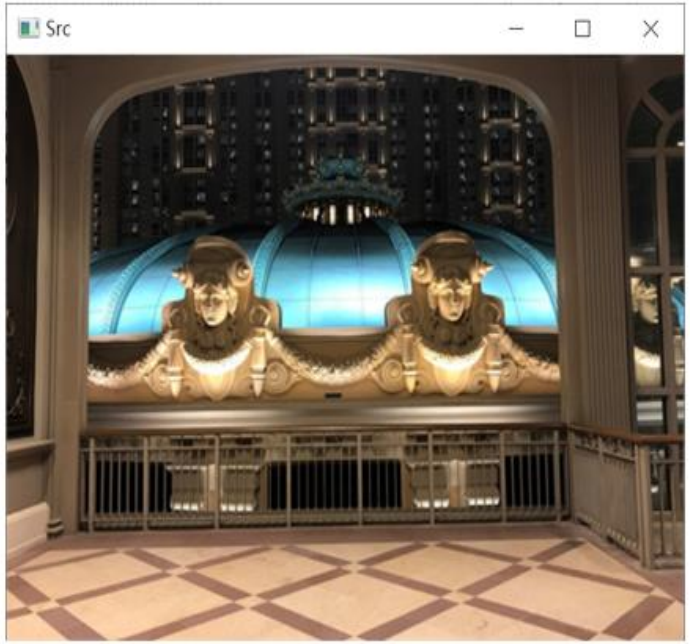
```
========== RESTART: D:\OpenCV_Python\ch19\ch19_7.py ==========
資料類型 = <class 'numpy.ndarray'>
資料外觀 = (256, 1)
資料大小 = 256
資料內容
[[   5.]
 [   5.]
 [   3.]
 [   7.]
 [  11.]
 [   9.]
 [   9.]
 [  17.]
 [  18.]
 [  25.]
```

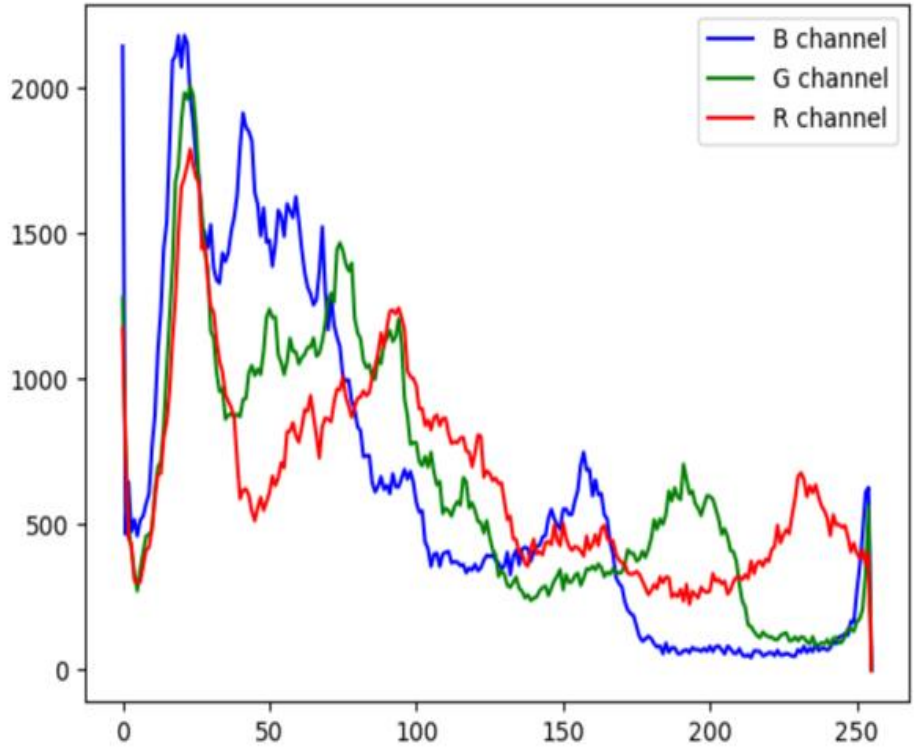
- 程式實例ch19_8.py：使用plot( )繪製snow.jpg影像的像素直方圖。

# 19-2-3：繪製彩色影像的直方圖

- 程式實例ch19_9.py：繪製macau.jpg影像的B、G、R通道的直方圖。

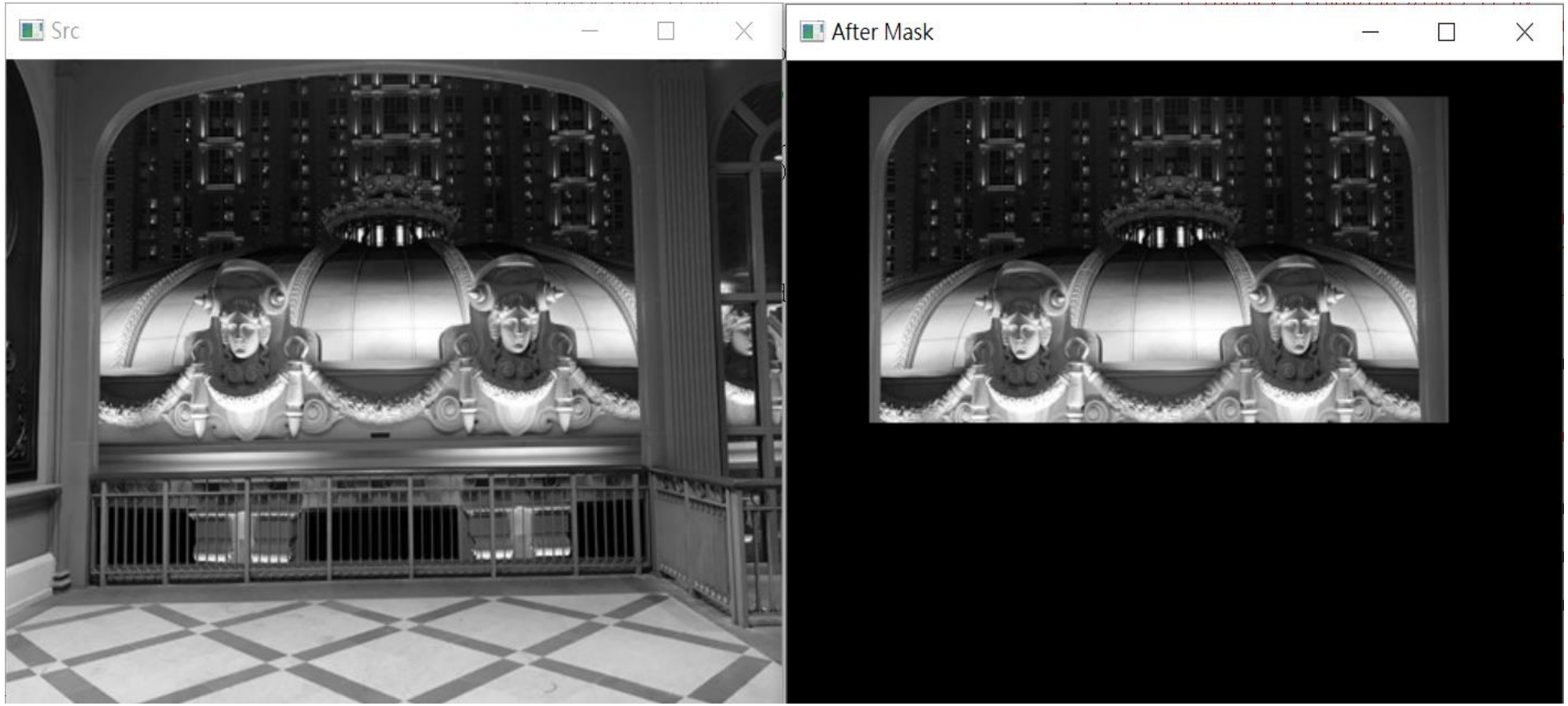


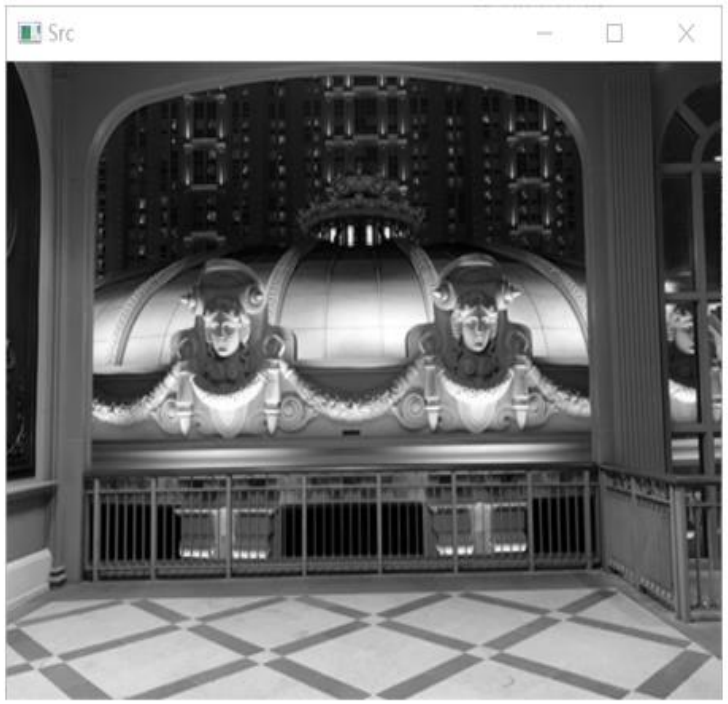
# 19-2-4：繪製遮罩的直方圖

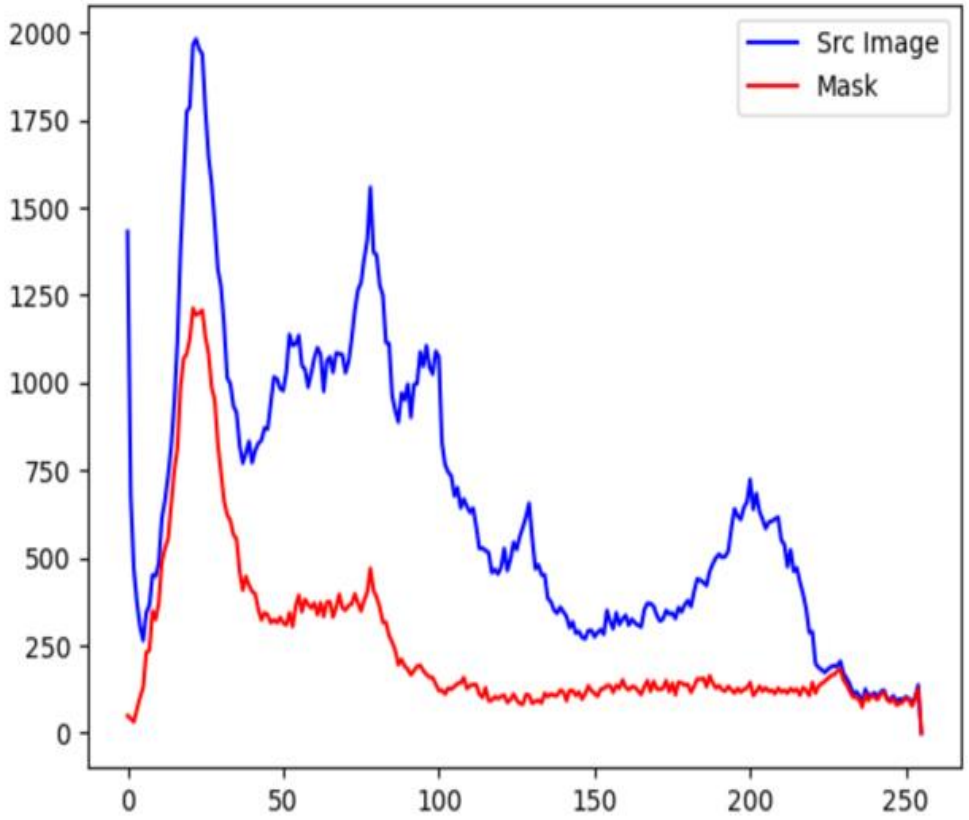
- 程式實例ch19_10.py：建立遮罩的方法，先建立一個影像區塊，然後在此影像區塊建立遮罩。

- 程式實例ch19_11.py：擴充ch19_10.py，在macau.jpg影像內建立遮罩，然後觀察執行結果。

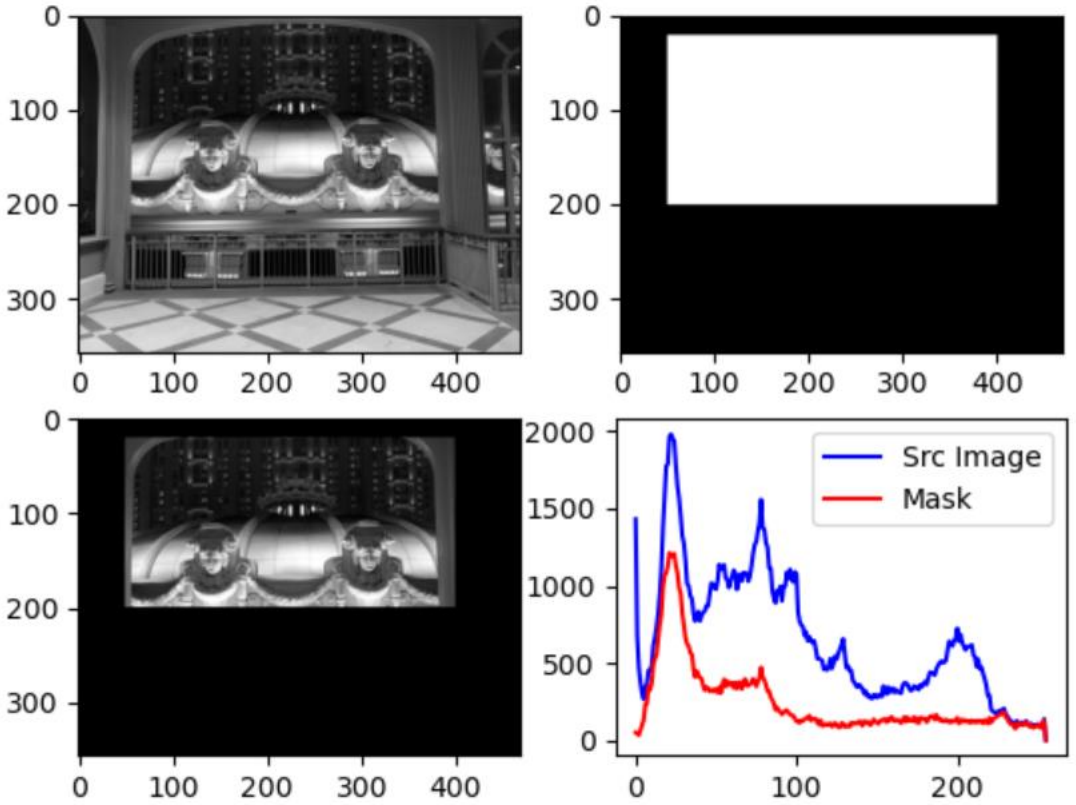


- 程式實例ch19_12.py：為整個影像和遮罩區的影像建立像素值的直方圖。

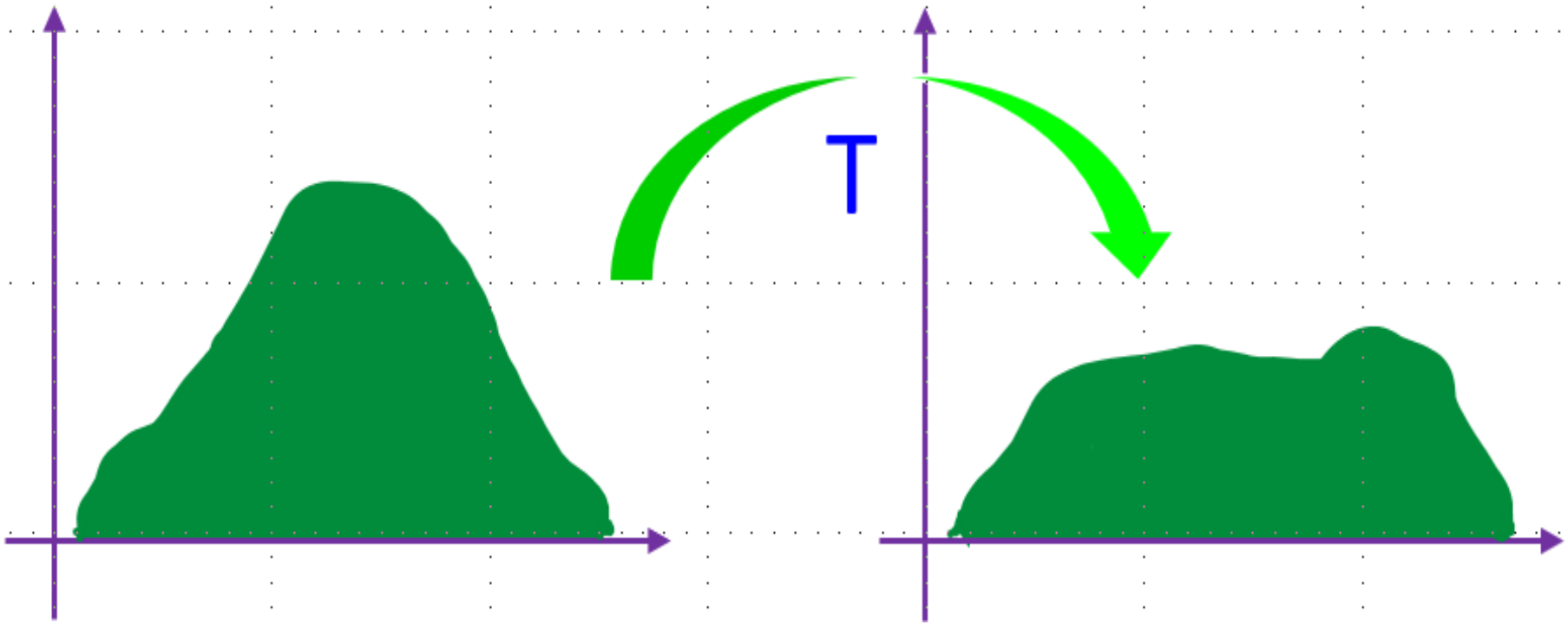


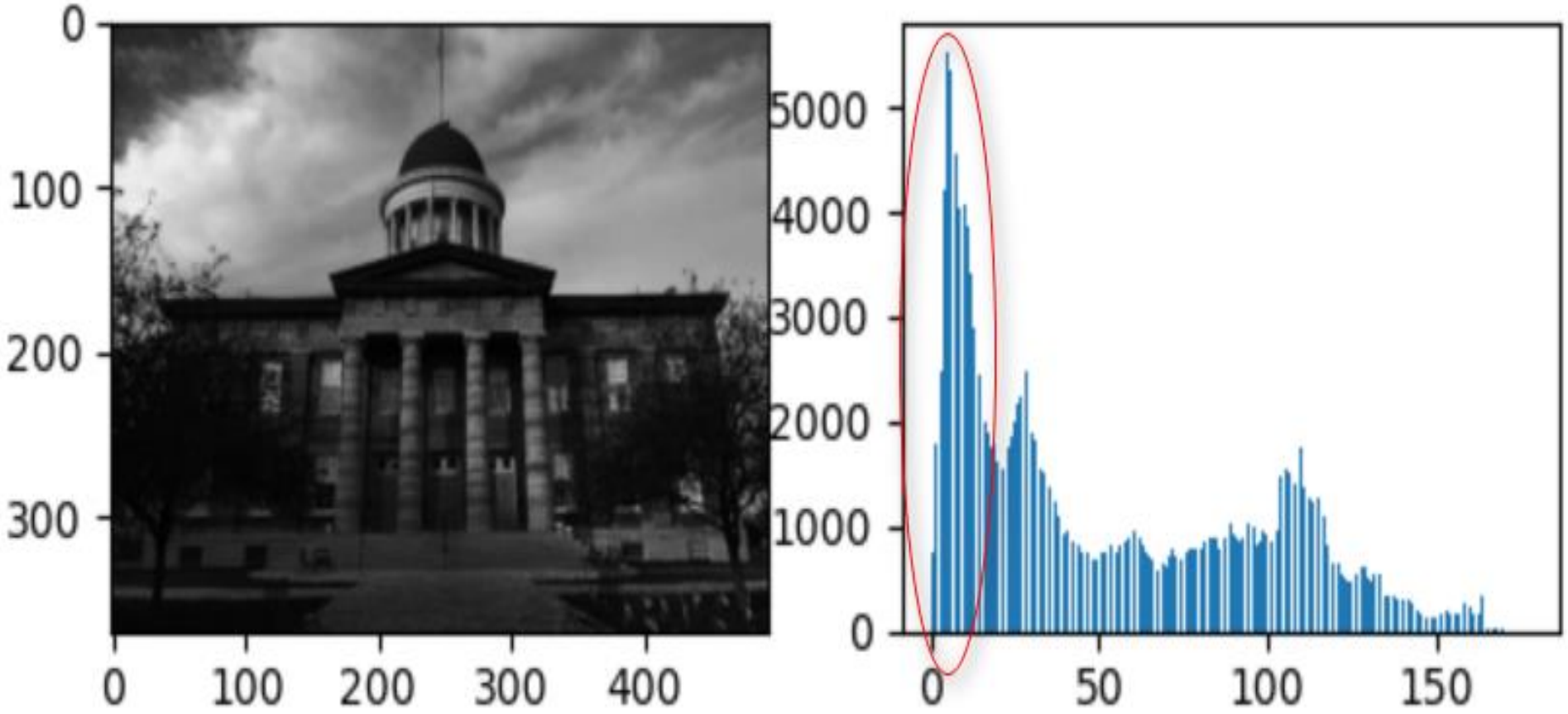
- 程式實例ch19_13.py：將ch19_11.py和ch19_2.py整合到一張圖表，這個程式主要是使用subplot( )函數，讀者可以參考第14和15行的說明。



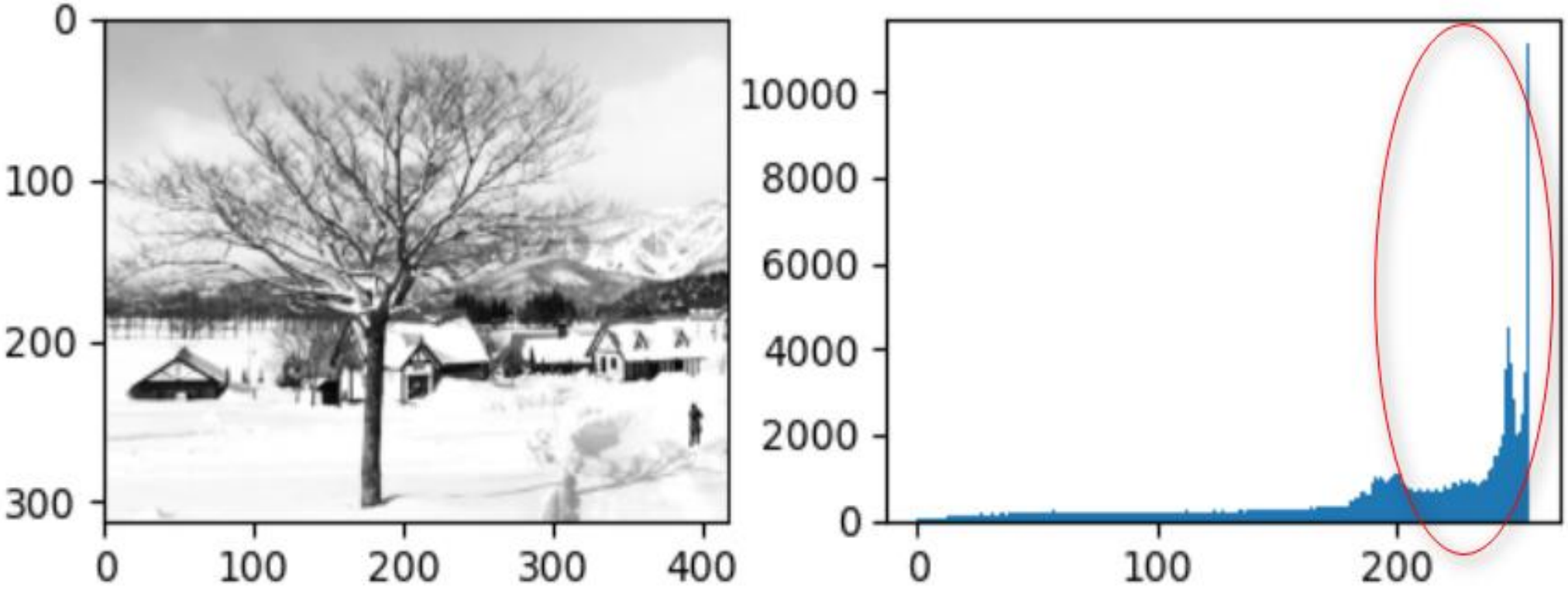
# 19-3：直方圖均衡化



- 通常過暗或過亮的圖往往是灰階值過度集中某一區域的結果，下列是過暗的影像與直方圖結果。



- 下列是過亮的影像與直方圖結果。



# 19-3-1：直方圖均衡化演算法

- 直方圖均衡化有2個步驟：
- 　1：計算累積的直方圖數據。
- 　2：將累積的直方圖執行區間轉換。

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 2 | 1 | 0 |
| 1 | 1 | 1 | 2 | 1 |
| 3 | 5 | 0 | 0 | 4 |
| 0 | 7 | 7 | 3 | 5 |
| 2 | 6 | 4 | 6 | 3 |

- 上述灰階值統計數據如下：

| 灰階值級 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 像素個數 | 6 | 5 | 3 | 3 | 2 | 2 | 2 | 2 |

- 將上述表格歸一化，可以得到下列結果。

| 灰階值級 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 像素個數 | 6 | 5 | 3 | 3 | 2 | 2 | 2 | 2 |
| 出現機率 | 6/25 | 5/25 | 3/25 | 3/25 | 2/25 | 2/25 | 2/25 | 2/25 |

- 使用小數點列出機率，結果如下：

| 灰階值級 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 像素個數 | 6 | 5 | 3 | 3 | 2 | 2 | 2 | 2 |
| 出現機率 | 0.24 | 0.2 | 0.12 | 0.12 | 0.08 | 0.08 | 0.08 | 0.08 |

- 計算累計機率，結果如下：

| 灰階值級 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 像素個數 | 6 | 5 | 3 | 3 | 2 | 2 | 2 | 2 |
| 出現機率 | 0.24 | 0.2 | 0.12 | 0.12 | 0.08 | 0.08 | 0.08 | 0.08 |
| 累計機率 | 0.24 | 0.44 | 0.56 | 0.68 | 0.76 | 0.84 | 0.92 | 1.00 |

- 接著有兩種均衡化的方法，如下：

❑在原有範圍執行均衡化。

❑在更廣泛的範圍執行均衡化。

## ❑ 在原有範圍執行均衡化

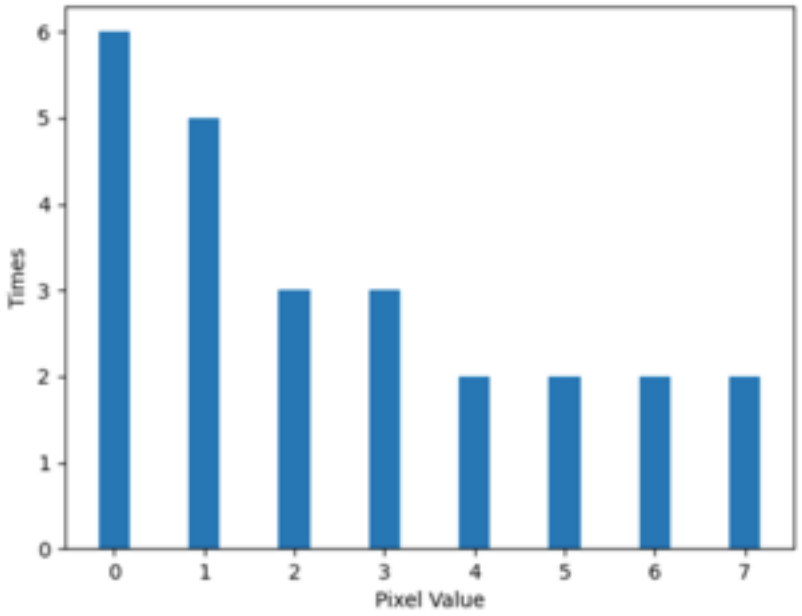
- 計算方式是用最大灰階值級，此例是7，乘以累積機率可以得到最新的灰階值級，可以使用四捨五入，最後可以得到下列結果。

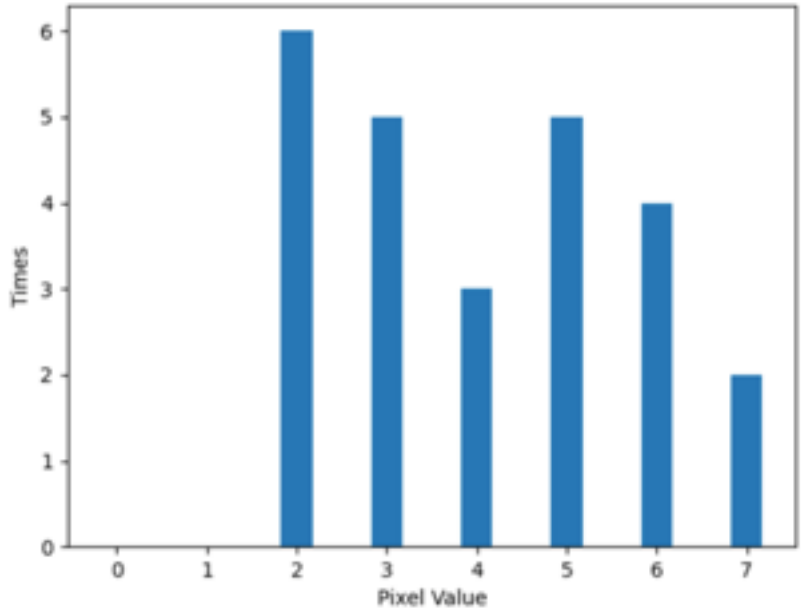
| 灰階值級 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 像素個數 | 6 | 5 | 3 | 3 | 2 | 2 | 2 | 2 |
| 出現機率 | 0.24 | 0.2 | 0.12 | 0.12 | 0.08 | 0.08 | 0.08 | 0.08 |
| 累計機率 | 0.24 | 0.44 | 0.56 | 0.68 | 0.76 | 0.84 | 0.92 | 1.00 |
| 新灰值級 | 2 | 3 | 4 | 5 | 5 | 6 | 6 | 7 |

- 上述新灰值級就是均衡化的結果，重新整理我們可以得到下列結果。

| 灰階值級 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 像素個數 | 0 | 0 | 6 | 5 | 3 | 5 | 4 | 2 |

- 下列是直方圖的比較圖，下方左圖是原始影像，下方有圖是均衡化結果。

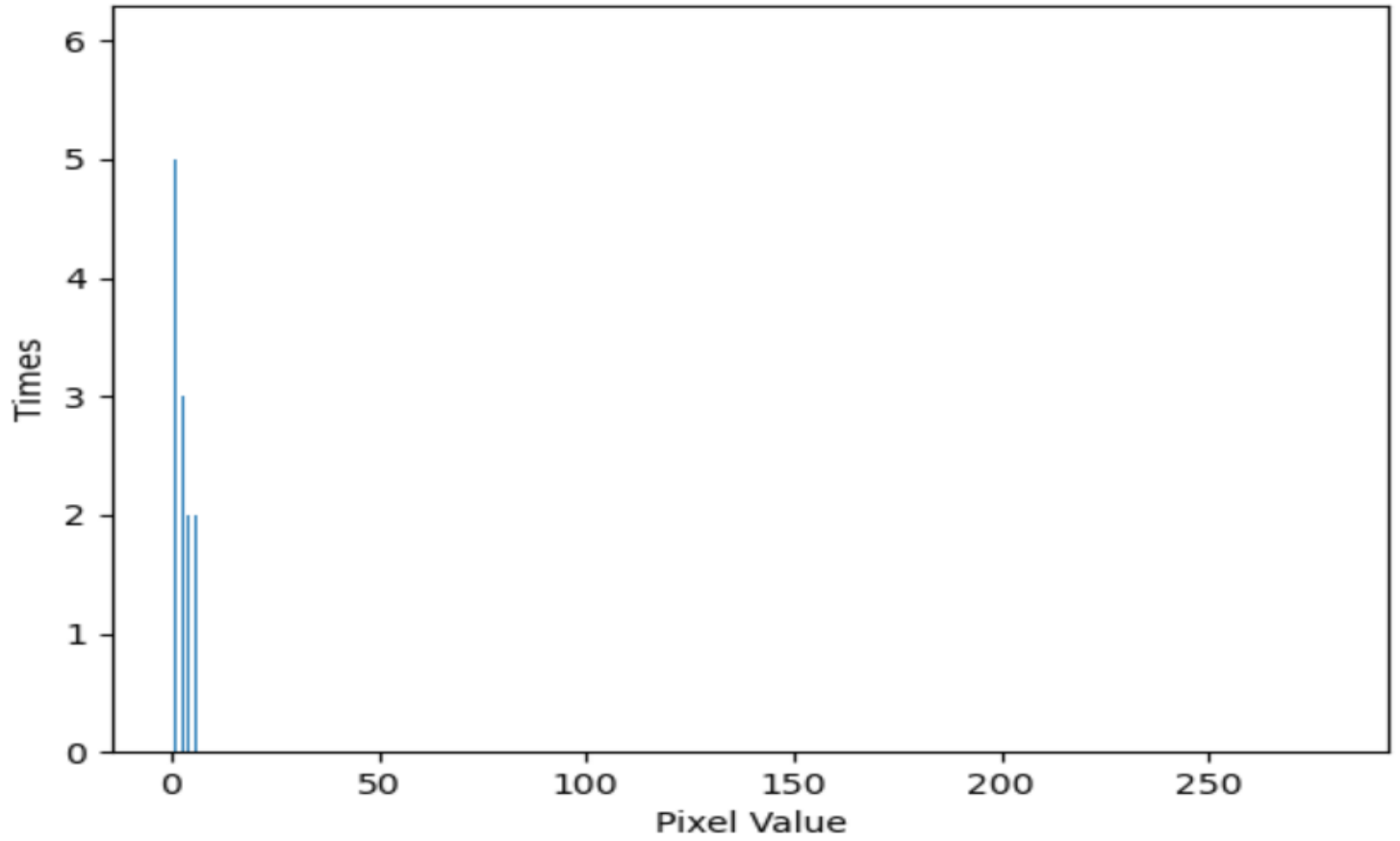


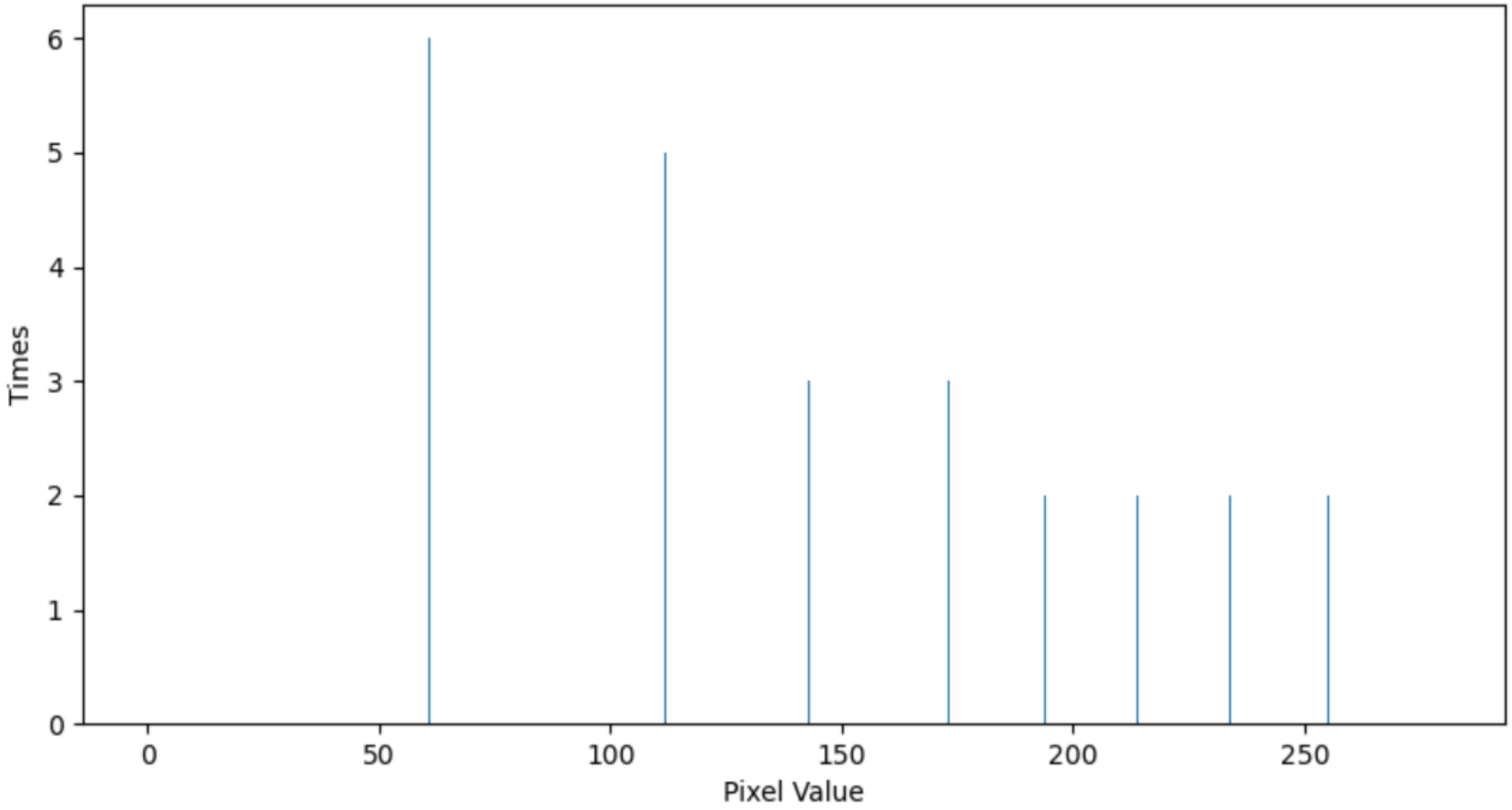
## ❑在更廣泛的範圍執行均衡化

| 灰階值級 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 像素個數 | 6 | 5 | 3 | 3 | 2 | 2 | 2 | 2 |
| 出現機率 | 0.24 | 0.2 | 0.12 | 0.12 | 0.08 | 0.08 | 0.08 | 0.08 |
| 累計機率 | 0.24 | 0.44 | 0.56 | 0.68 | 0.76 | 0.84 | 0.92 | 1.00 |
| 新灰值級 | 61 | 112 | 143 | 173 | 194 | 214 | 234 | 255 |

- 下列是原先的直方圖。

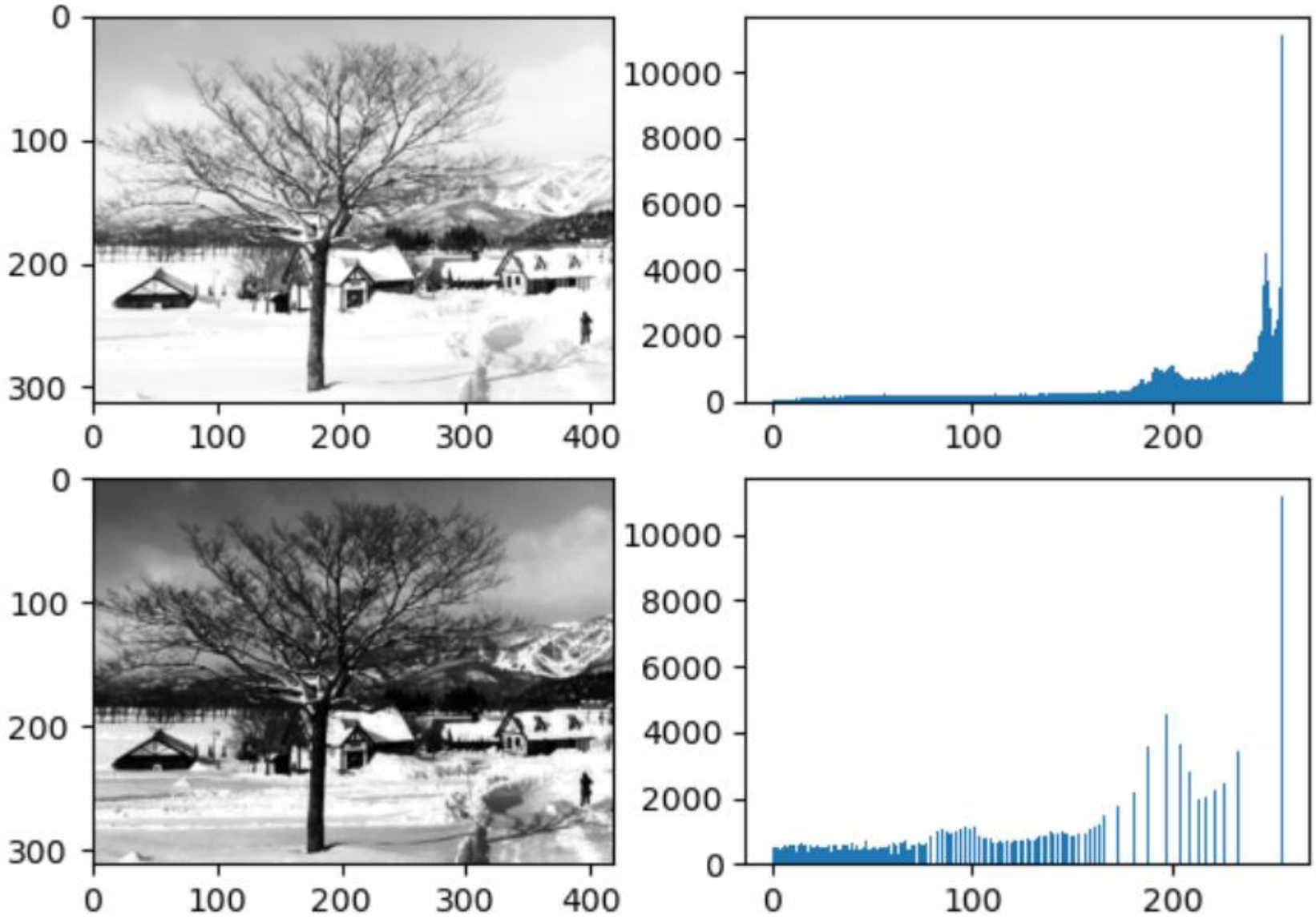


- 下列是更廣泛執行均衡化的結果。

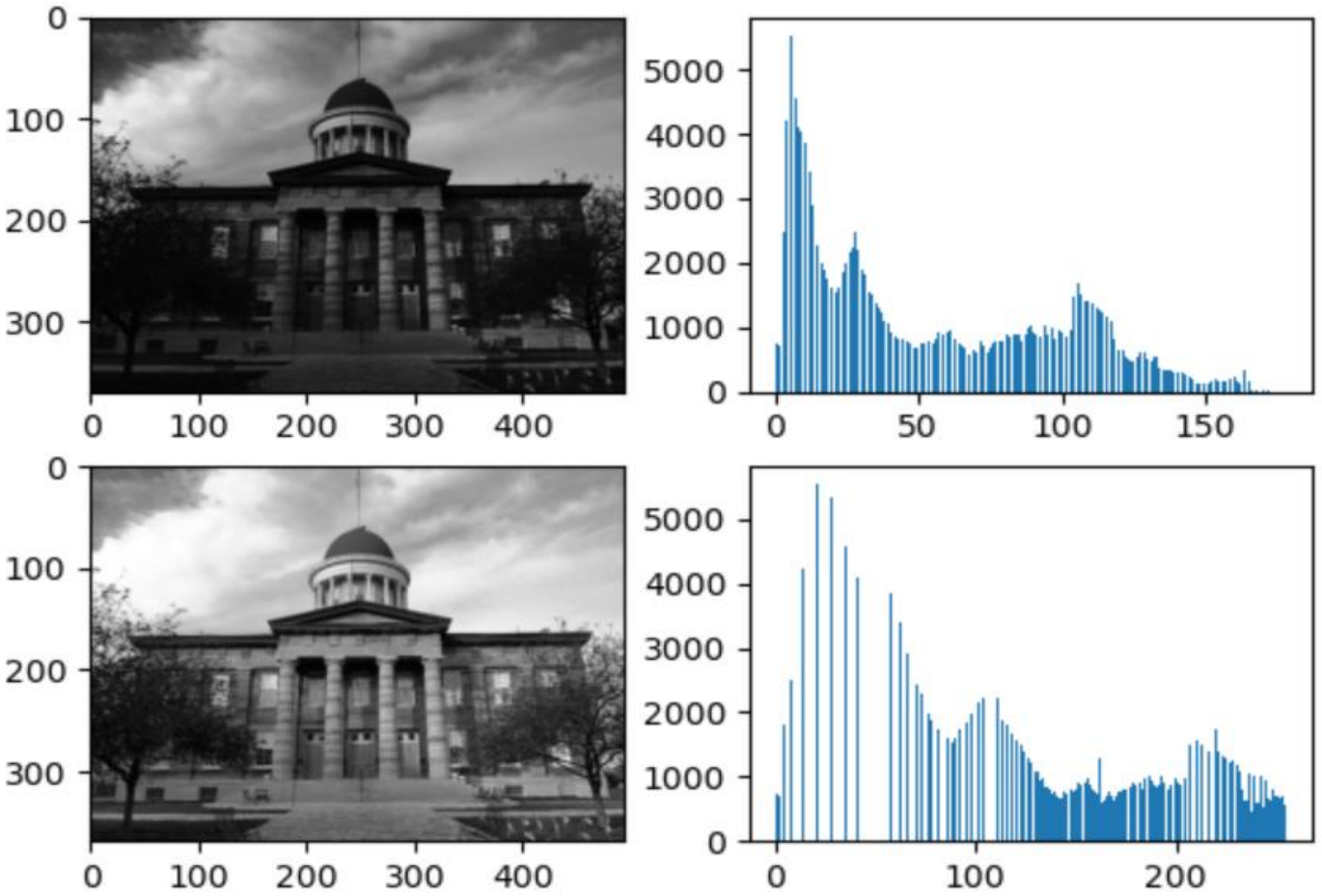


# 19-3-2：直方圖均衡化equalizeHist( )

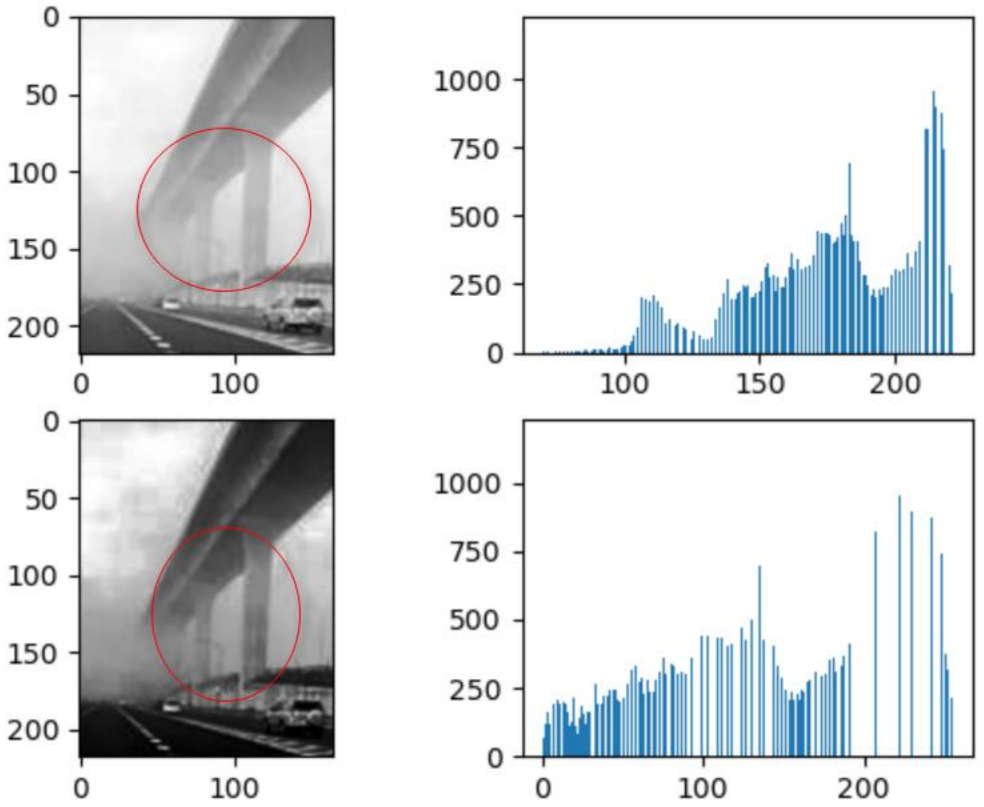
- dst = cv2.equalizeHist(src)
- 程式實例ch19_14.py：snow1.py是過度曝光太亮的影像，使用直方圖均衡化，同時列出執行結果。


0
100
200
300
0
100
200
300
400
10000
8000
6000
4000
2000
0
0
100
200

- 程式實例ch19_15.py：springfield.py是過暗的影像，使用直方圖均衡化，同時列出執行結果，這一實例只是將所讀取的snow1.jpg改成springfield.jpg檔案。

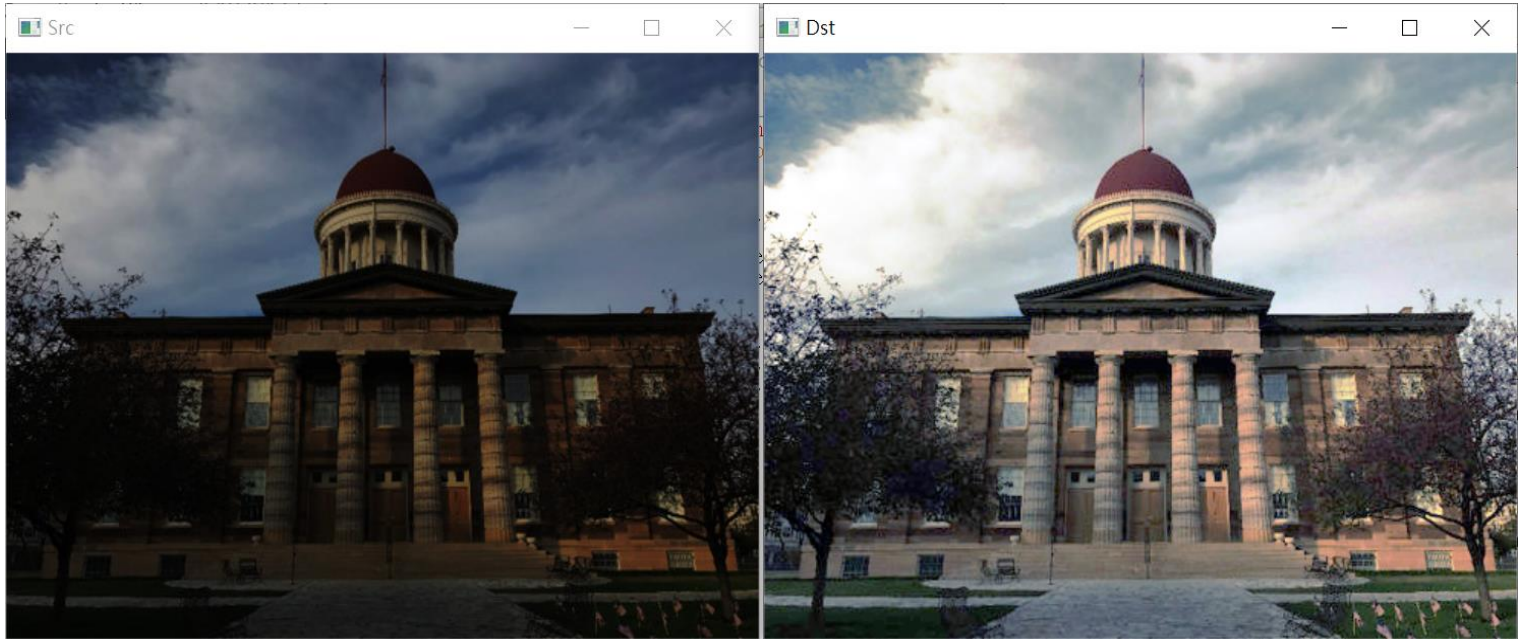


- 程式實例ch19_16.py：去除霧的實例。



# 19-3-3：直方圖均衡化應用在彩色影像

- 程式實例ch19_17.py：使用springfield.jpg，執行彩色影像的直方圖均衡化。

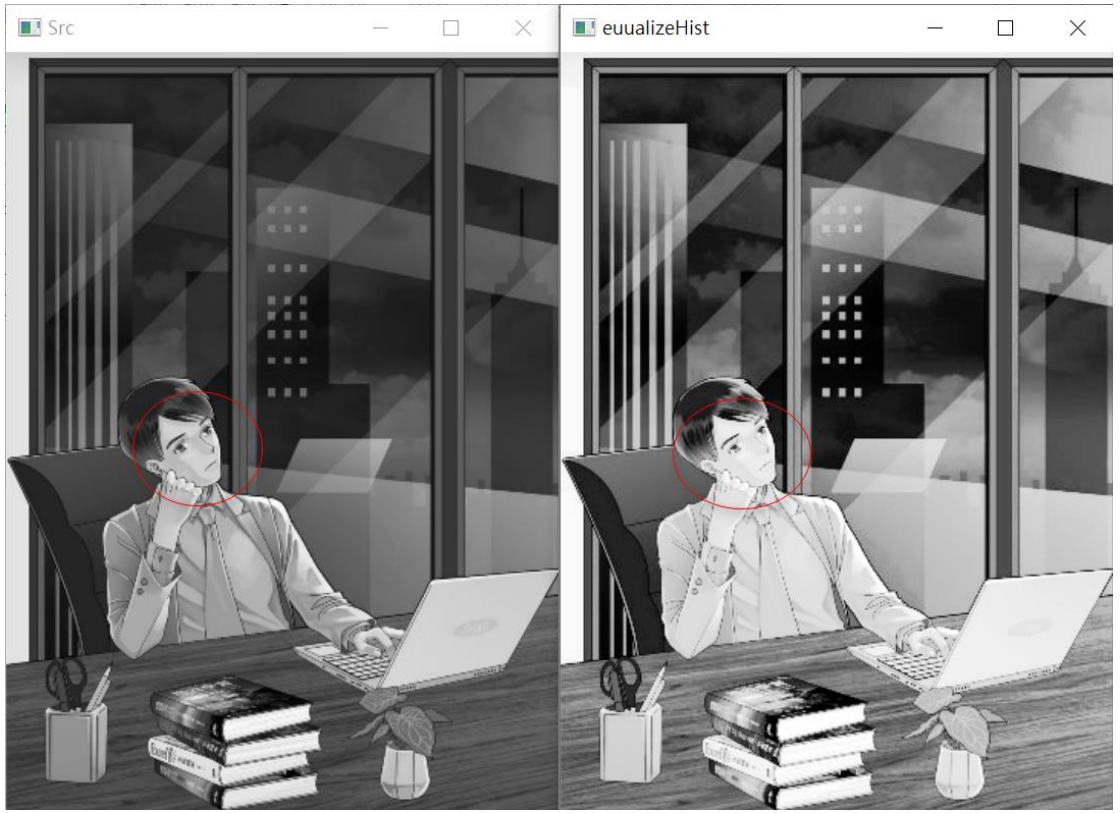


# 19-4：限制自適應直方圖均衡化方法

- 19-4-1：直方圖均衡化的優缺點

# 19-4-2：直方圖均衡化的缺點實例

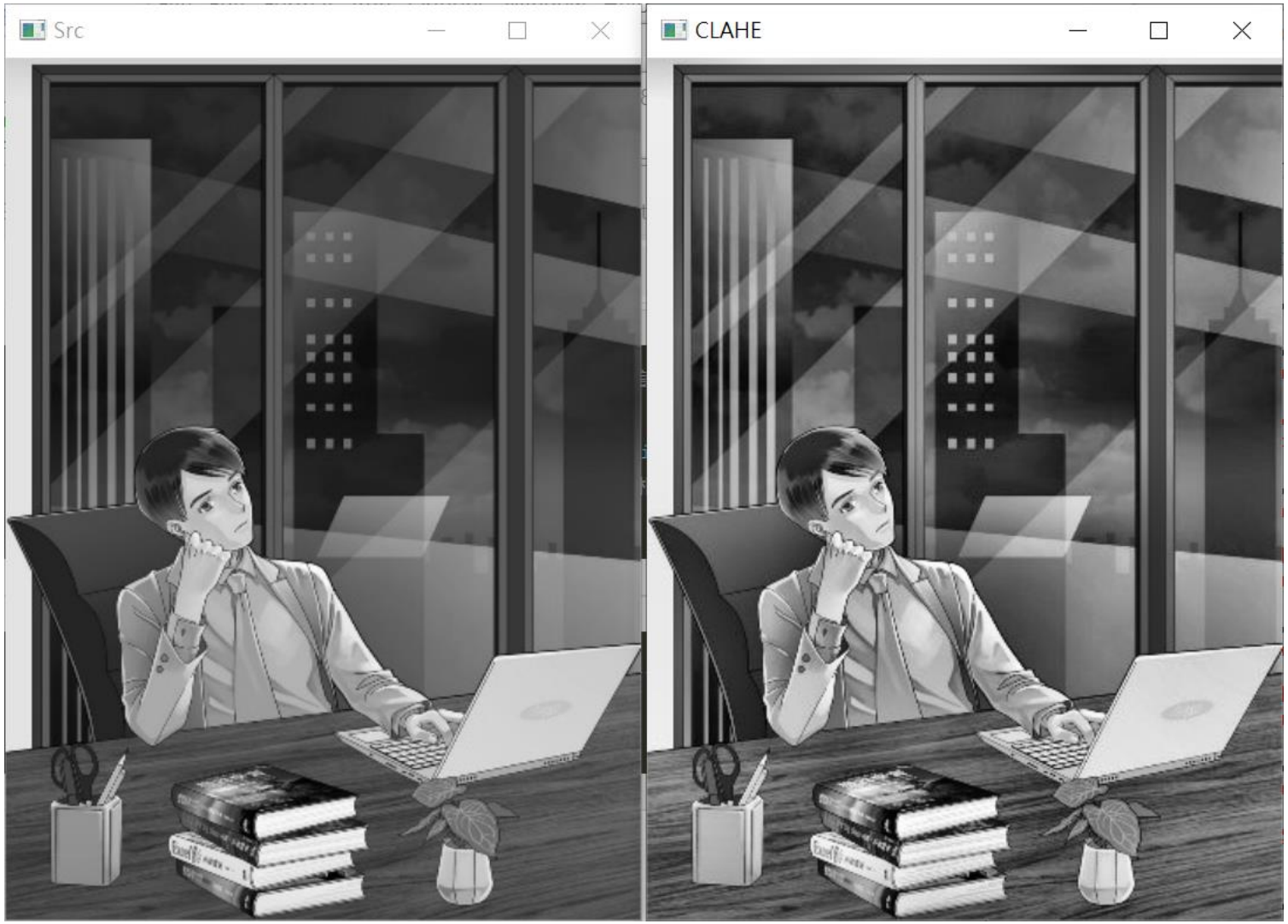
- 程式實例ch19_18.py：使用直方圖均衡化處理office.jpg。



# 19-4-3：自適應直方圖函數createCLAHE( )和apply( )函數

- clahe = cv2.createCLACHE(clipLimit, tileGridSize)
- dst = clahe.apply(src_gray)　　# src_gray是灰階影像物件

- 程式實例ch19_19.py：使用自適應直方圖函數，重新設計ch19_18.py。
-


Src
CLAHE