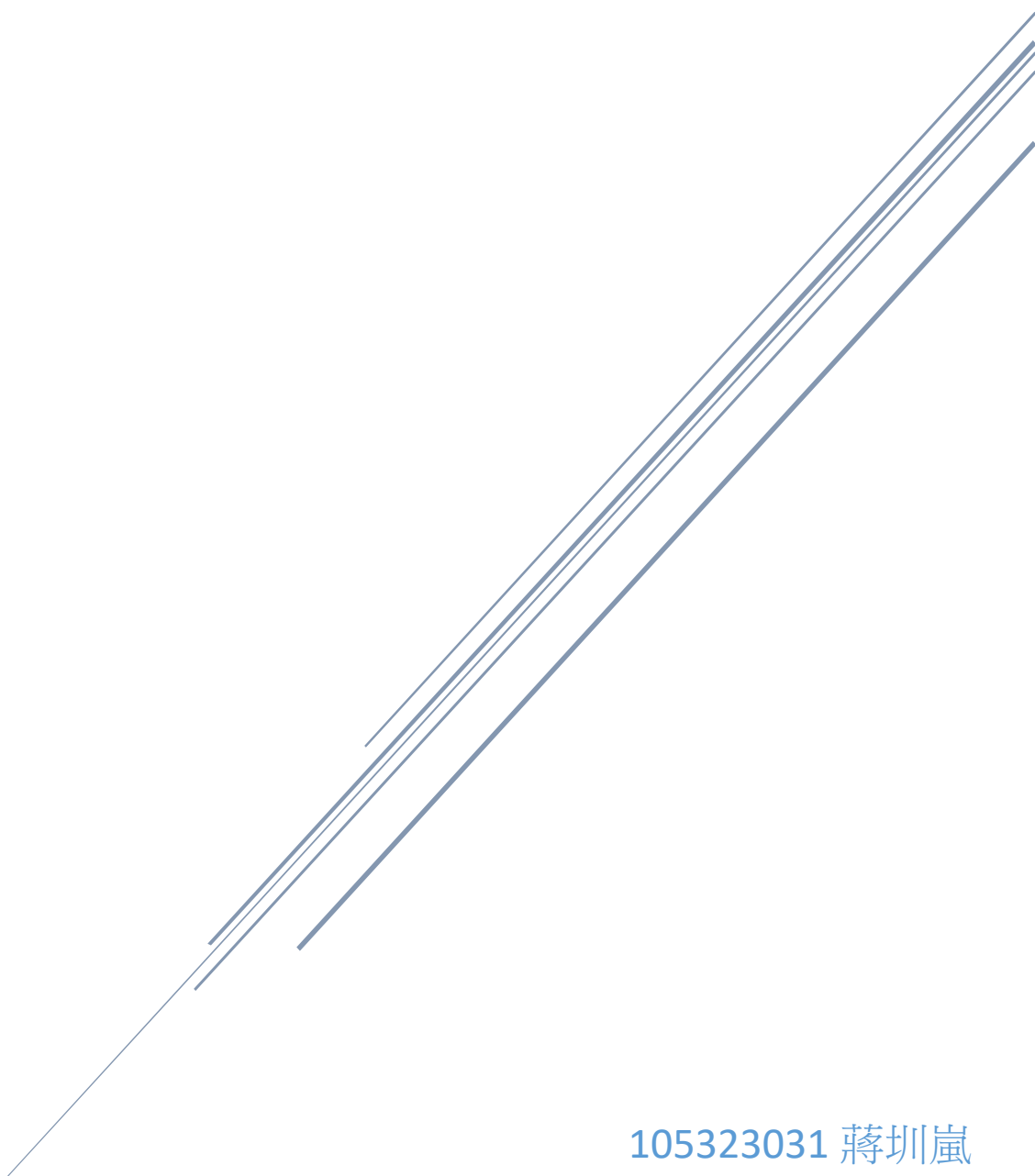


# 數位視訊技術



105323031 蔣圳嵐

# 原理:

Binarization: 使用 threshold 的方式，讓數值超過一定值就為另一個值，就可以造成只有兩個值的情況。

Histogram: 可以看出整張圖數值的分布狀況，從而去做影像處理。

# 程式說明:

Binarization:

```
1 import numpy as np
2 import cv2
3 import matplotlib.pyplot as plt
4
5 img = cv2.imread('Jet.bmp')
6 a, binarization = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)
7 cv2.imshow('Jet-bin', binarization)
8 cv2.waitKey(0)
9 cv2.destroyAllWindows()
```

最開始使用 opencv 的 imread 函式讀入檔案並放在名為 img 的名稱內，之後使用 threshold 的函式將讀入的影像 img 灰階值超過 127 的都變成 255，最後使用 imshow 把經過處理的影像顯示在螢幕上。

Histogram:

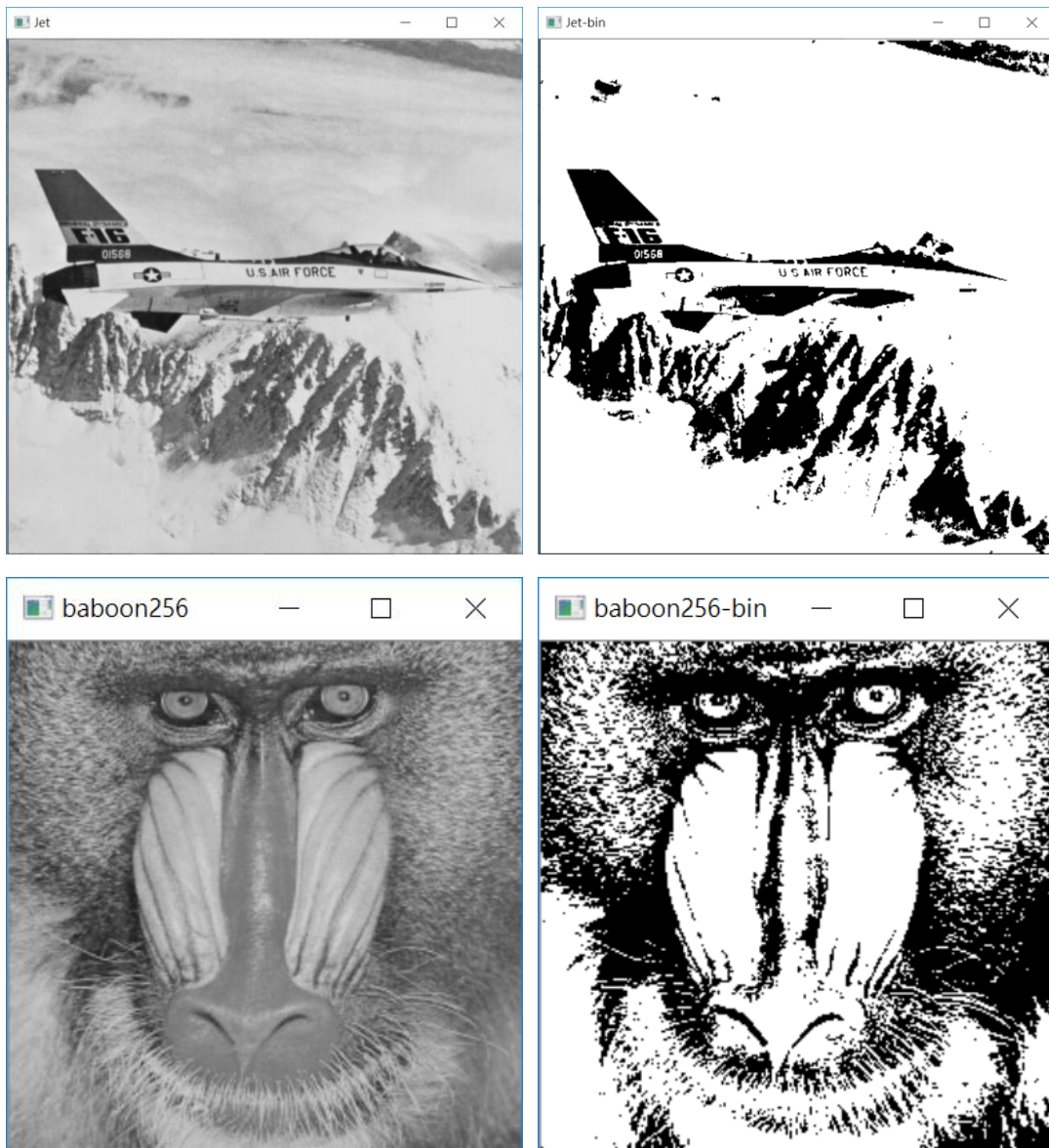
```
1 import numpy as np
2 import cv2
3 import matplotlib.pyplot as plt
4
5 img = cv2.imread('Jet.bmp')
6
7 hist = cv2.calcHist([img], [0], None, [256], [0, 256])
8 plt.plot(hist)
9 plt.show()
```

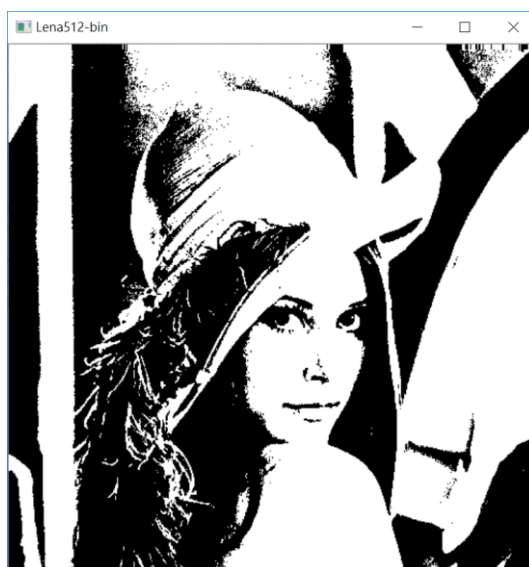
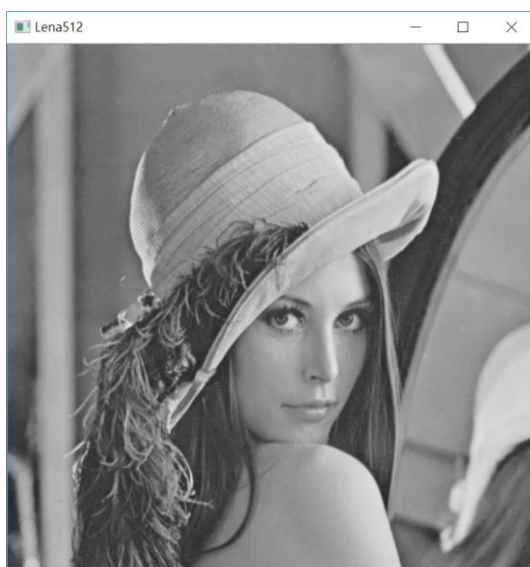
一樣先使用 imread 把檔案讀入，之後使用 opencv 內 calcHist 的函式，這是用來計算 histogram 的函式，算完將值丟給 hist，最後再將 hist 用 matplotlib.pyplot 函式內的 plot 畫出來。

註: `histogram` 函式有五個參數，第一個為輸入圖像，第二個為通道，如果有使用到彩圖就可以分為 RGB 三個通道，第三個為遮罩，第四個繪畫出來的直方圖要幾個區間，第五個為計算像素質的範圍通常為 $[0,256]$ 。

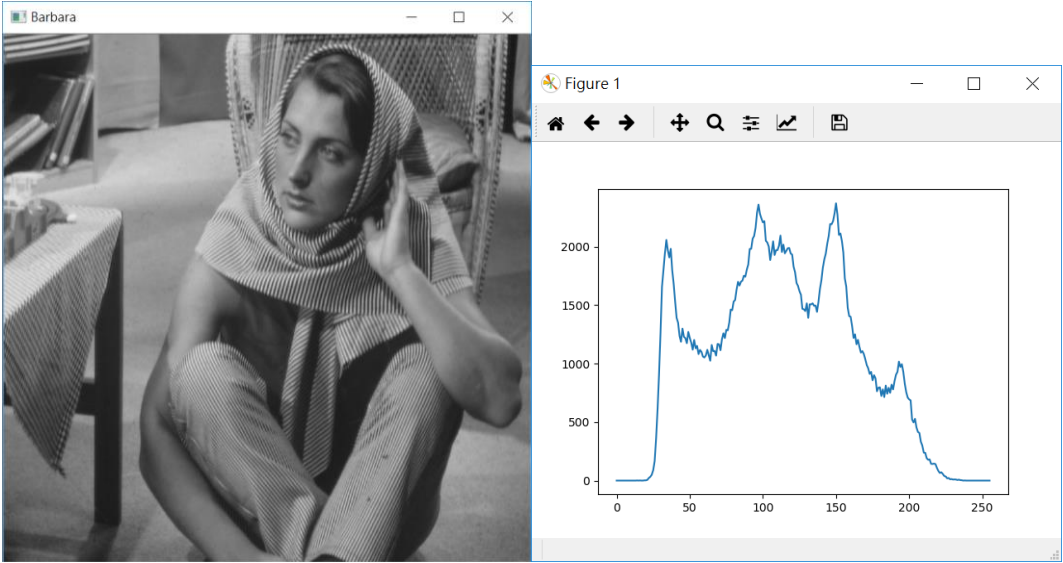
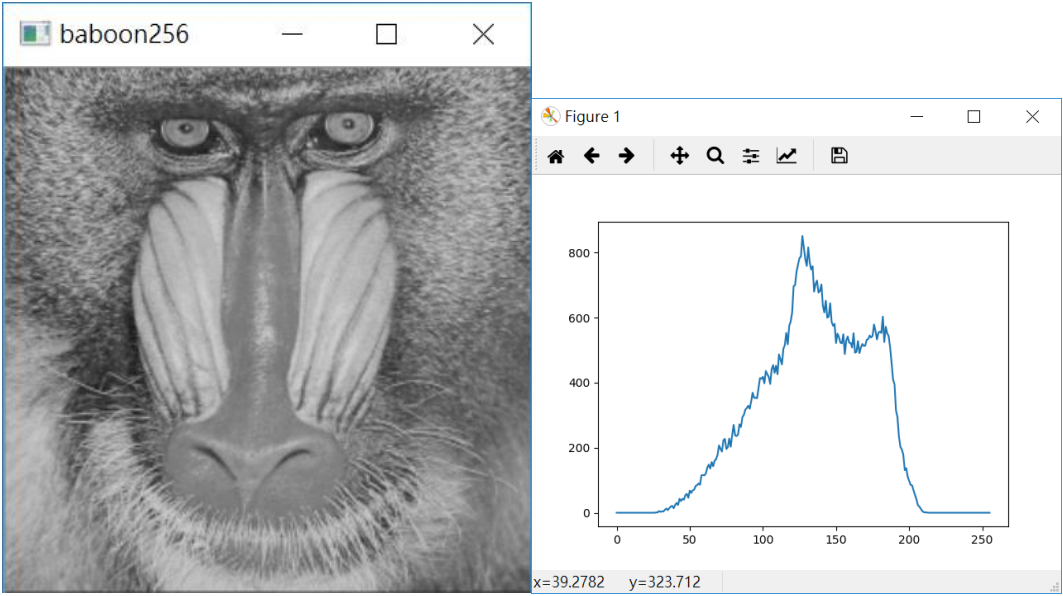
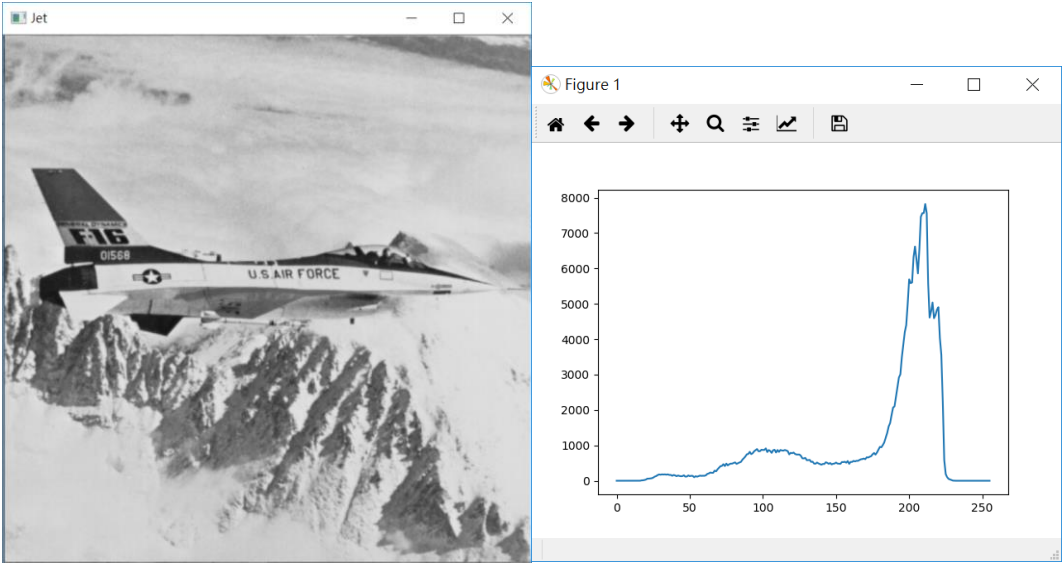
## 程式結果:

Binarization:

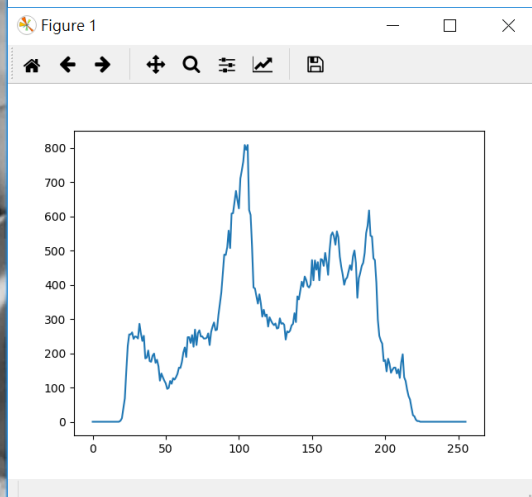
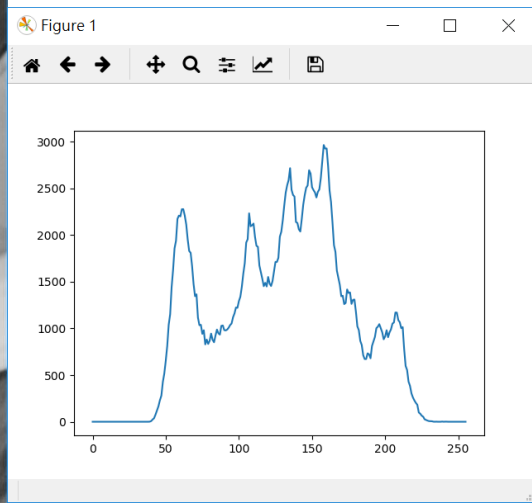
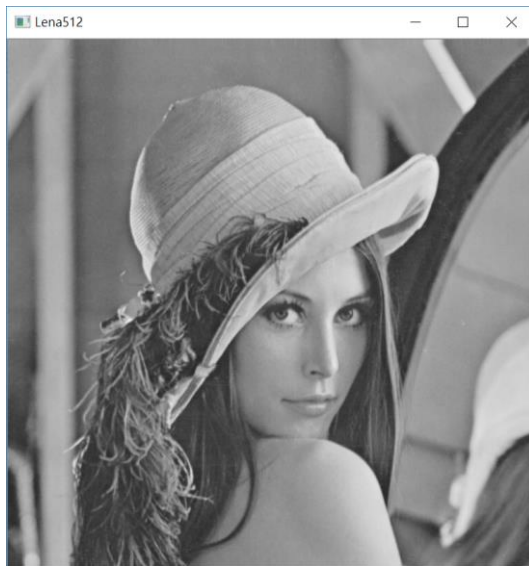




Histogram:







## 結論：

平常我很少使用 `python` 去打程式，這此因為老師不斷說 `python` 一行指令就解決了，於是就來嘗試。初次使用對於一些東西很不習慣，因為 `python` 是一個相對比較不嚴謹的語言，因此有些指令都含糊的打，但是習慣了以後反而覺得打起來相當輕鬆，一行指令就能解決相當多事情，是很方便的一個語言。

此次作業是做 `binarization` 和 `histogram`，前者早期的應用都在一些文件上面，讓背景能夠更白，字更明顯。而我覺得這個應用也能濾掉一些不想看到的東西，結合 `AI` 就能讓深度學習更專注地學我們餵給他的資料。而後者的應用我想應該也只有老師上課舉的那個例子就是當整張照片過暗的時候，先使用 `histogram` 把數值的分布畫出來之後再做正規化，就可以得到一個看起來比較舒適的照片。