影像處理HW1

直方圖(Histogram) 與 直方圖均化 (Histogram Equalization)

黑白影像直方圖

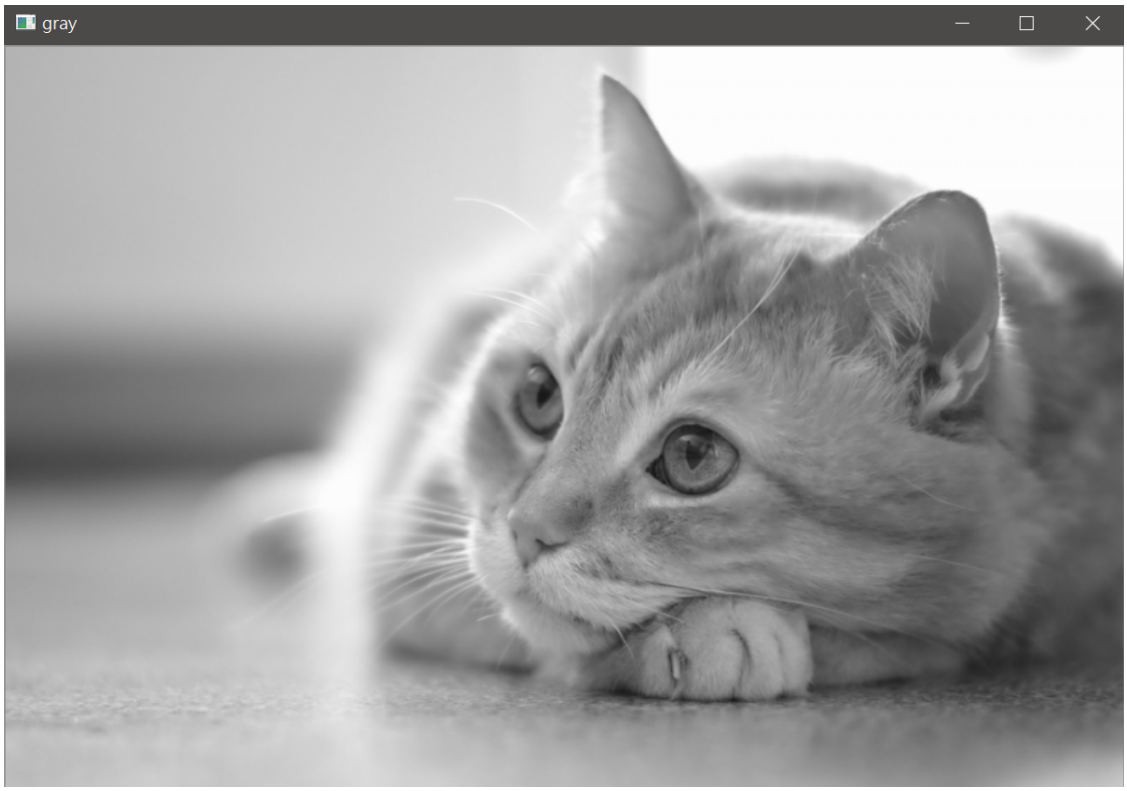
- 使用 calcHist 套件

```
"""
# 計算直方圖每個 bin 的數值
# cv2.calcHist(影像, 通道, 遮罩, 區間數量, 數值範圍)
hist = cv2.calcHist([gray], [0], None, [256], [0, 256])
"""
```

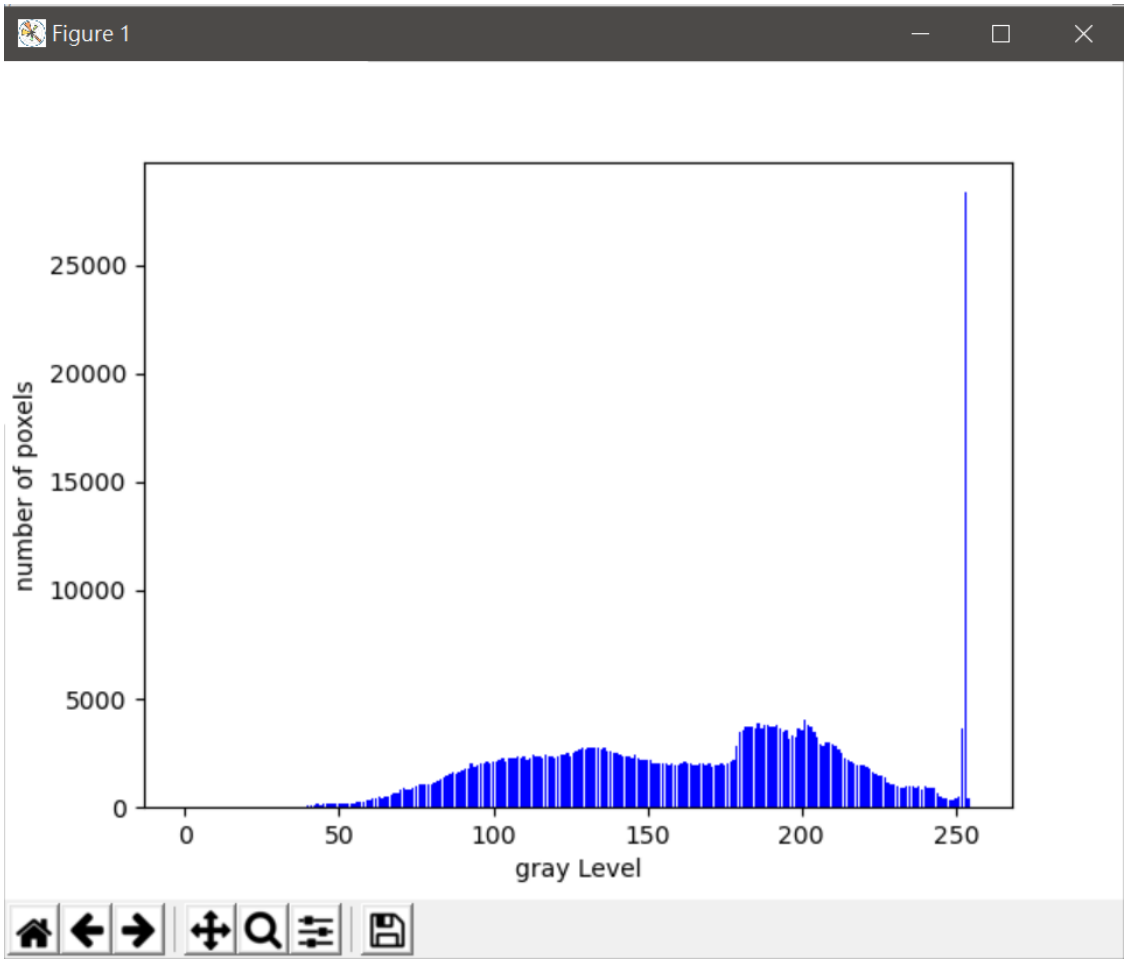
- 不使用 calcHist 套件

```
import sys
import numpy as np
import cv2
import matplotlib.pyplot as plt


img = cv2.imread("cat.jpg", cv2.IMREAD_GRAYSCALE)
# cv2.imshow(winname,mat) 顯示影像
cv2.imshow("gray", img)
# cv2.Waitkey([delay]) 等待按鍵
cv2.waitKey()

# 建立一維256位置的0陣列
canvs = np.zeros((256), np.int)
# img 像素維度 -> 一維 reshape(-1)
image = np.reshape(img, (-1))
# 計算向素值的數量
for num in image:
    canvs[num] += 1
print(canvs)
# 直方圖
x = range(256)
plt.xlabel('gray Level')
plt.ylabel("number of poxels")
# plt.plot(x, canvs)
plt.bar(x, canvs, align='center', color='b')
plt.show()
```

- 原圖



- 直方圖



黑白影像直方圖均化

- 累積分布函數（cdf）：把直方圖累加 / 最後累加之像素值

- 公式:

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{cdf_{max} - cdf_{min}} \times (L-1)\right)$$

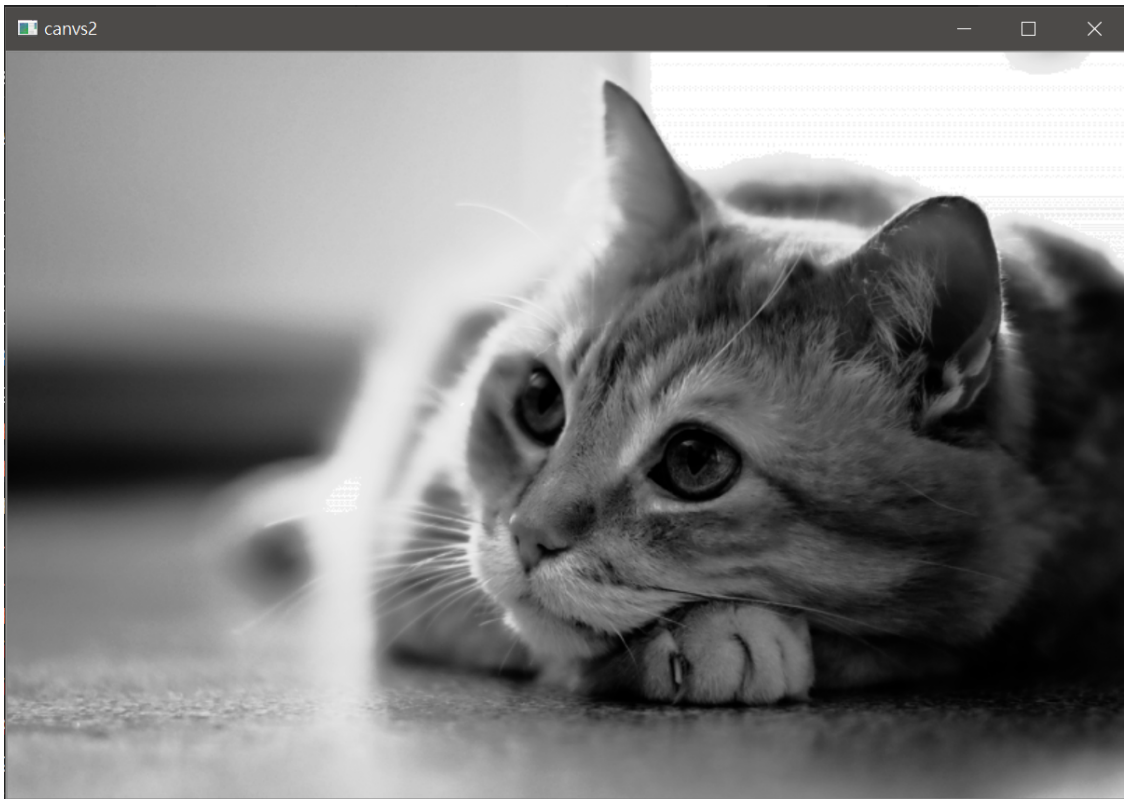
ex.灰階為78的像素的累積分布函數為46，均衡化後，灰階值變化為：

$$h(78) = \text{round}\left(\frac{46-1}{63} \times 255\right) = \text{round}\left(0.714286 \times 255\right) = 182$$

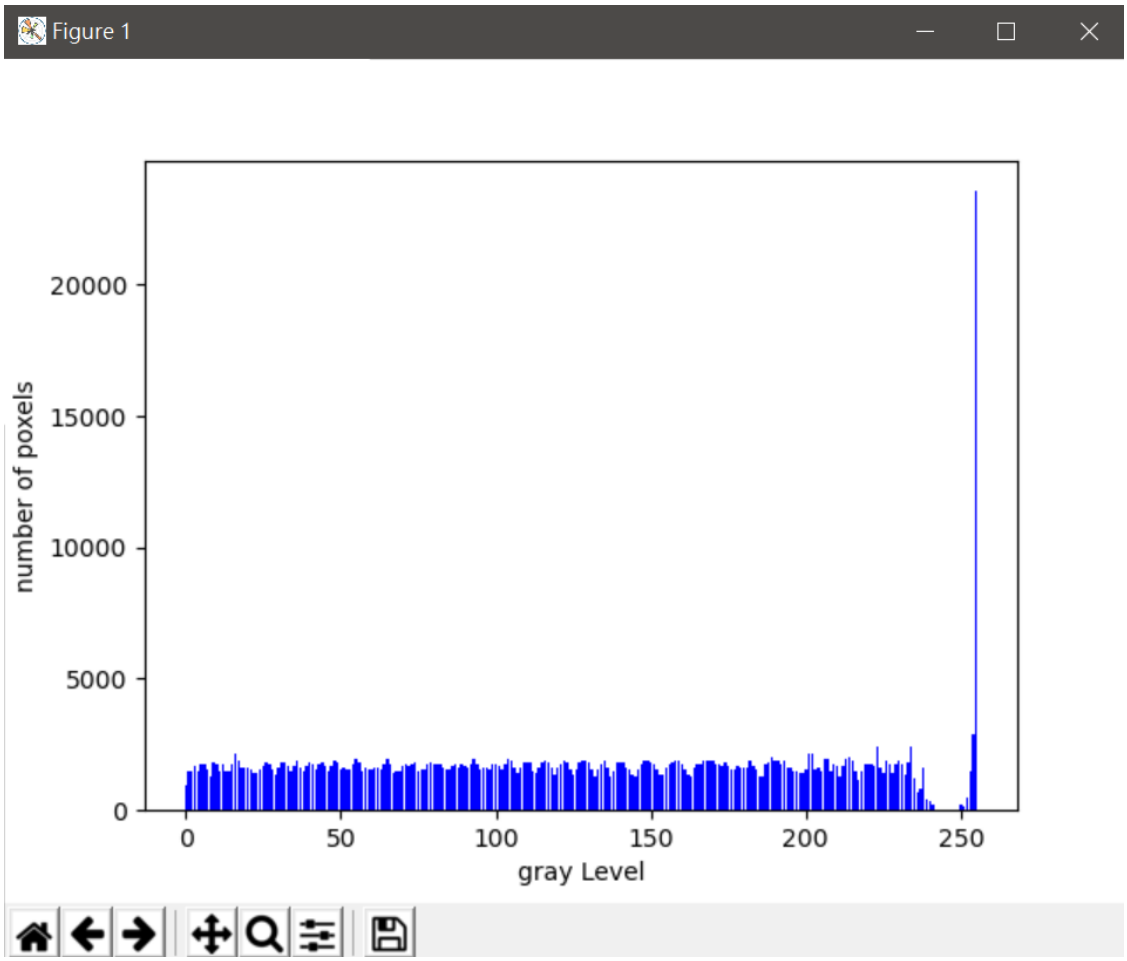
```
import sys
import numpy as np
import cv2
import matplotlib.pyplot as plt

img = cv2.imread("cat.jpg",cv2.IMREAD_GRAYSCALE)

canvs = np.zeros((256), np.int)
image = np.reshape(img,(-1))

for num in image:
    canvs[num] += 1

# ============================================================
tmp = np.zeros((256), np.int)

i = 0
tmp[0] = canvs[0]
for i in range(1,256): # 1-255
    tmp[i] = tmp[i-1] + canvs[i]
# print(tmp[255]) # 累加all
# print(tmp)
tmp1 = list(tmp)
while 0 in tmp1:
    tmp1.remove(0)
# tmp2[np.argmin(tmp)] = np.max(tmp)

# cdf = np.zeros((256), np.float)
cdf = tmp / tmp[255]
# print(cdf)
cdf_min = min(cdf)
cdf_max = max(cdf)
canvs2 = np.zeros((534,800),np.uint8)
for row in range(img.shape[0]):
    for col in range(img.shape[1]):
        # for img[row,col] all pixel(534*800)'s grayLevel value -> equalization
        canvs2[row,col] = np.round((cdf[img[row,col]]-cdf_min) / (cdf_max-
cdf_min) * (256-1))

cv2.imshow('canvs2', canvs2)
cv2.imwrite('cat2.jpg', canvs2)
cv2.waitKey(0)

img2 = cv2.imread("cat2.jpg", cv2.IMREAD_GRAYSCALE)
```

```
# cv2.imshow("gray",img2)

cv2.waitKey()

canvs3 = np.zeros((256), np.int)
image2 = np.reshape(img2, (-1))
# 計算向素值的數量
for num in image2:
    canvs3[num] += 1
print(canvs3)

x = range(256)
plt.xlabel('gray Level')
plt.ylabel("number of poxels")
# plt.plot(x, canvs3)
plt.bar(x, canvs3, align='center', color='b')
plt.show()
```

- 均化圖



- 均化值方圖

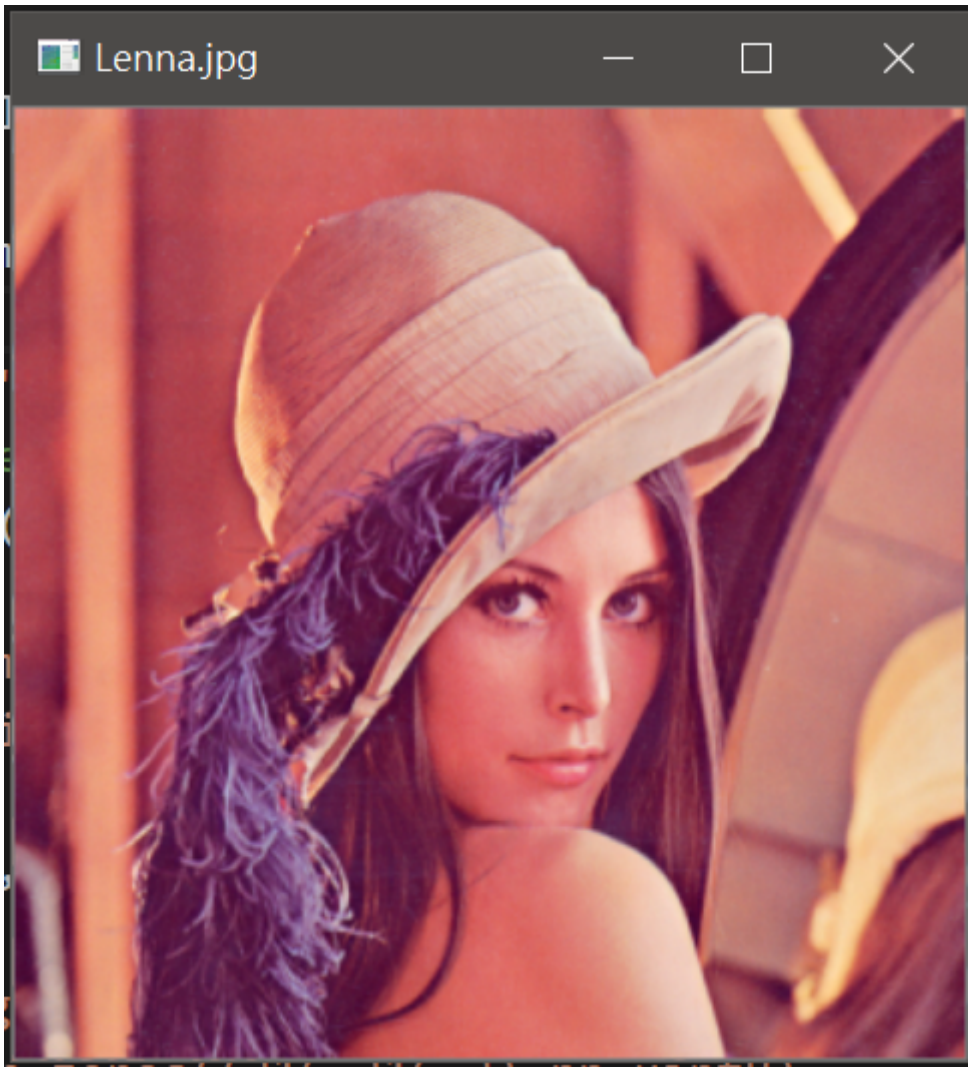


彩色影像直方圖(三張)

```
import sys
import numpy as np
import cv2
import matplotlib.pyplot as plt

img = cv2.imread("Lenna.jpg")

"""
cv2.split(img) 拆分影像通道
cv2.merge([img_b,img_g,img_r]) 通道合併
"""
img_b,img_g,img_r=cv2.split(img)

""" 顯示 img b 通道
canvs_b = np.zeros((316,316,3),np.uint8)
canvs_b[:,:,0] = img_b
cv2.imwrite('canvs_b.jpg',canvs_b)
image = cv2.imread('canvs_b.jpg')
cv2.imshow("canvs_b",image)
cv2.waitKey(0)
"""

""" 顯示 img g 通道
canvs_g = np.zeros((534,800,3),np.uint8)
canvs_g[:,:,0] = img_g
cv2.imwrite('canvs_g.jpg',canvs_g)
image = cv2.imread('canvs_g.jpg')
```

```
cv2.imshow("canvs_g",image)
cv2.waitKey(0)
"""

""" 顯示 img r 通道
canvs_r = np.zeros((534,800,3),np.uint8)
canvs_r[:,:,0] = img_r
cv2.imwrite('canvs_r.jpg',canvs_r)
image = cv2.imread('canvs_b.jpg')
cv2.imshow("canvs_r",image)
cv2.waitKey(0)
"""

canvs_b = np.zeros((256), np.int)
canvs_g = np.zeros((256), np.int)
canvs_r = np.zeros((256), np.int)

image_b = np.reshape(img_b,(-1))
image_g = np.reshape(img_g,(-1))
image_r = np.reshape(img_r,(-1))

for num in image_b:
    canvs_b[num] += 1

for num in image_g:
    canvs_g[num] += 1

for num in image_r:
    canvs_r[num] += 1

x = range(256)
plt.xlabel('gray Level')
plt.ylabel("number of poxels")
# plt.plot(x, canvs_b,'b')
plt.bar(x, canvs_b, align='center', color='b')
plt.show()
# plt.plot(x, canvs_g,'g')
plt.bar(x, canvs_g, align='center', color='g')
plt.show()
# plt.plot(x, canvs_r,'r')
plt.bar(x, canvs_b, align='center', color='r')
plt.show()
```

- 原圖



- 值方圖

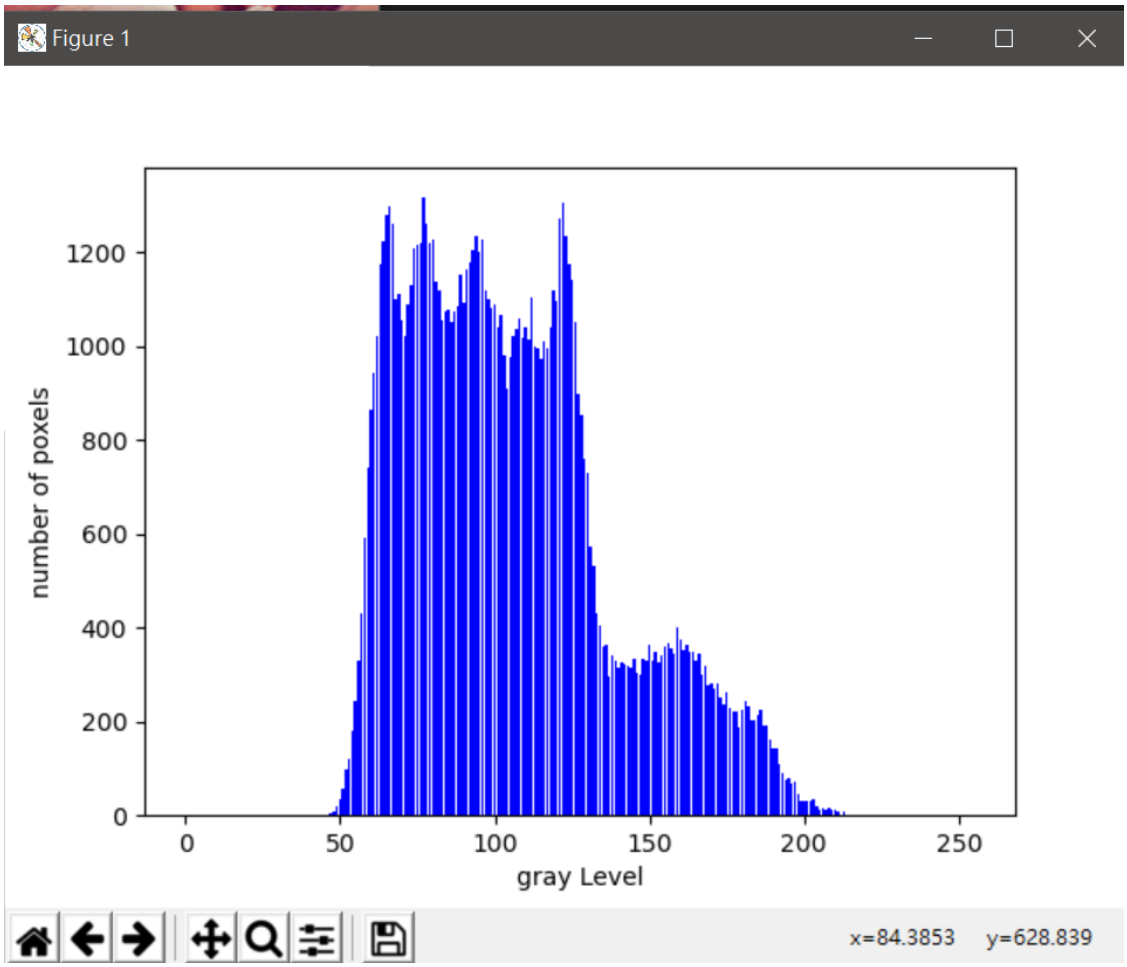
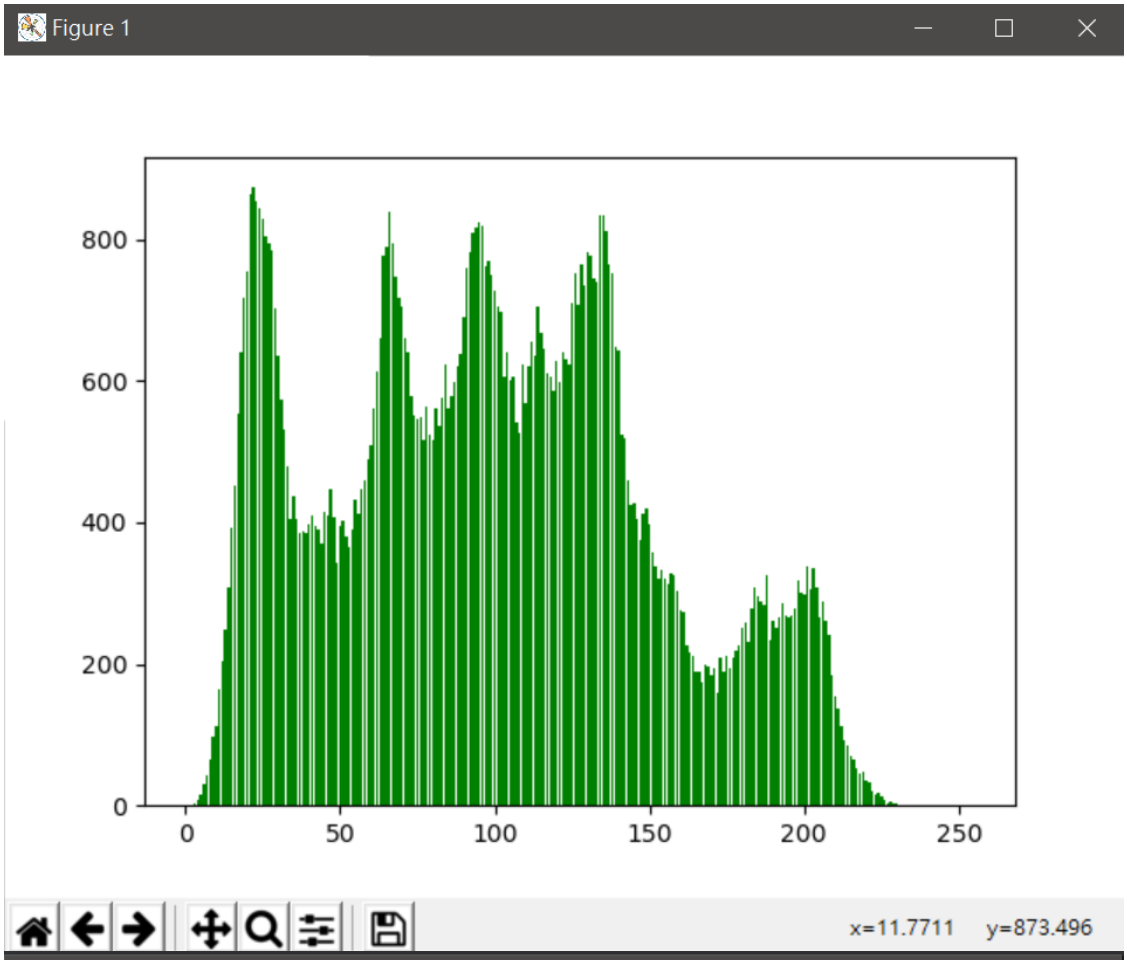
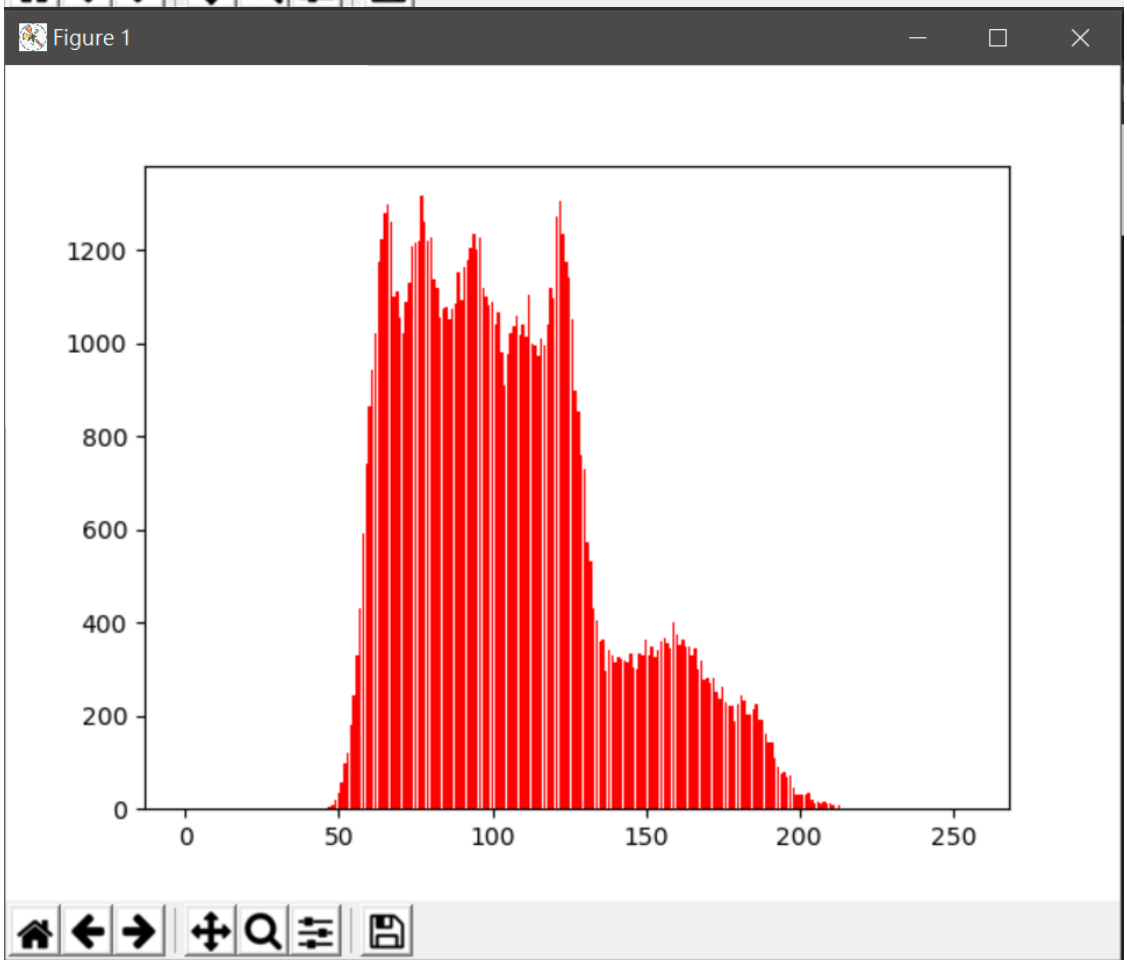


Figure 1
1200
1000
800
600
400
200
0
number of poxels
0
50
100
150
200
250
gray Level
x=84.3853
y=628.839





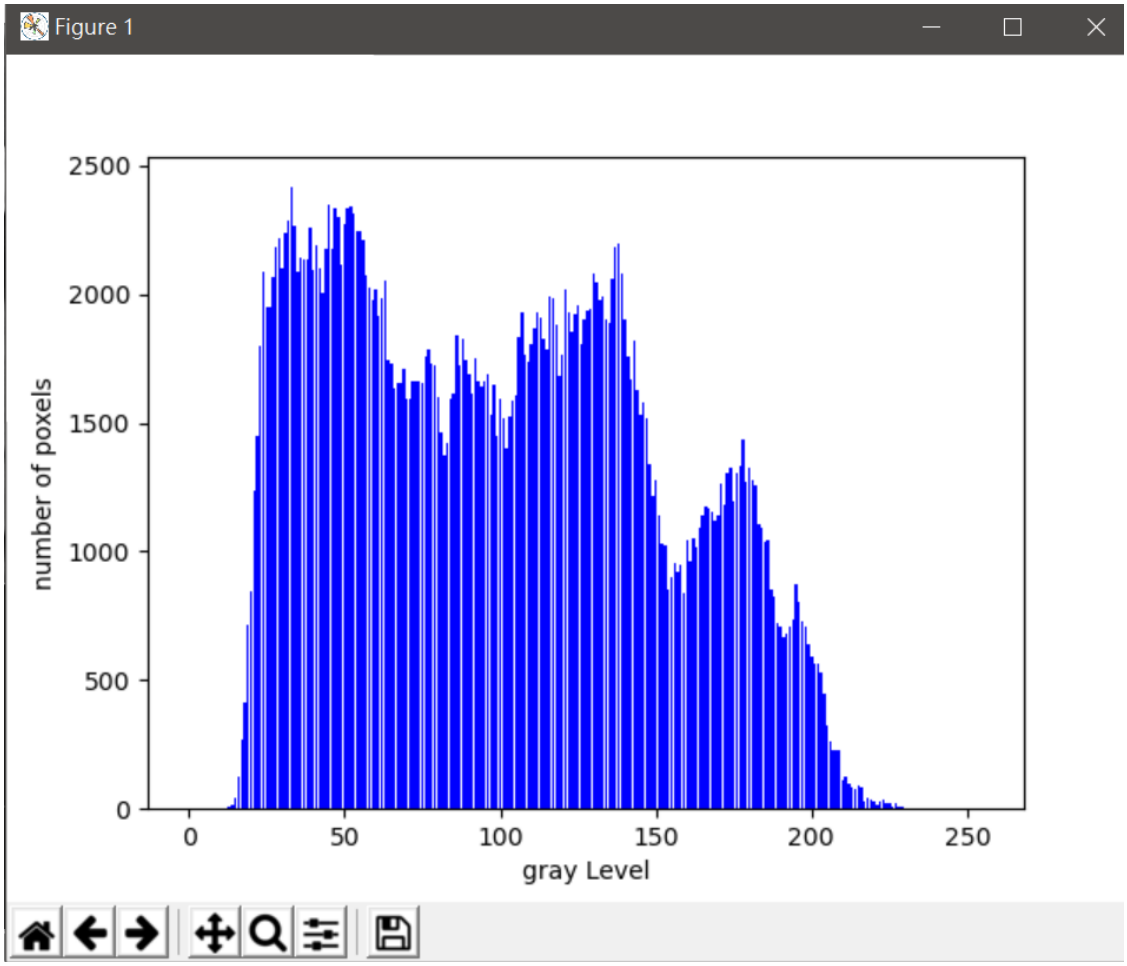
彩色影像直方圖分別均化並合併出結果

- B 通道 均化

```
import sys
import numpy as np
import cv2
import matplotlib.pyplot as plt

img = cv2.imread("Lenna.jpg")
img_b,img_g,img_r=cv2.split(img)

canvs_b = np.zeros((256), np.int)
image_b = np.reshape(img,(-1))

for num in image_b:
    canvs_b[num] += 1

# ============================================================
tmp_b = np.zeros((256), np.int)

i = 0
tmp_b[0] = canvs_b[0]
for i in range(1,256):
    tmp_b[i] = tmp_b[i-1] + canvs_b[i]

tmp1_b = list(tmp_b)
while 0 in tmp1_b:
    tmp1_b.remove(0)

cdf_b = tmp_b / tmp_b[255]
cdf_min_b = min(cdf_b)
cdf_max_b = max(cdf_b)
canvs2_b = np.zeros((316,316),np.uint8)
for row in range(img_b.shape[0]):
    for col in range(img_b.shape[1]):
        # for img[row,col] all pixel(534*800)'s grayLevel value -> equalization
        canvs2_b[row,col] = np.round((cdf_b[img_b[row,col]]-cdf_min_b) /
(cdf_max_b-cdf_min_b) * (256-1))

cv2.imshow('canvs2_b',canvs2_b)
cv2.imwrite('Lenna_b.jpg',canvs2_b)
cv2.waitKey(0)

img2 = cv2.imread("Lenna_b.jpg")
img2_b,img2_g,img2_r=cv2.split(img2)
# cv2.imshow("gray",img2)

# cv2.waitKey()

canvs3_b = np.zeros((256),np.int)
image2_b = np.reshape(img2,(-1))
# 計算向素值的數量
for num in image2_b:
    canvs3_b[num] += 1
print(canvs3_b)


# 直方圖
x = range(256)
plt.xlabel('gray Level')
plt.ylabel("number of poxels")
```

```
plt.plot(x, canvs3_b)
plt.show()
```

- 值方圖均化



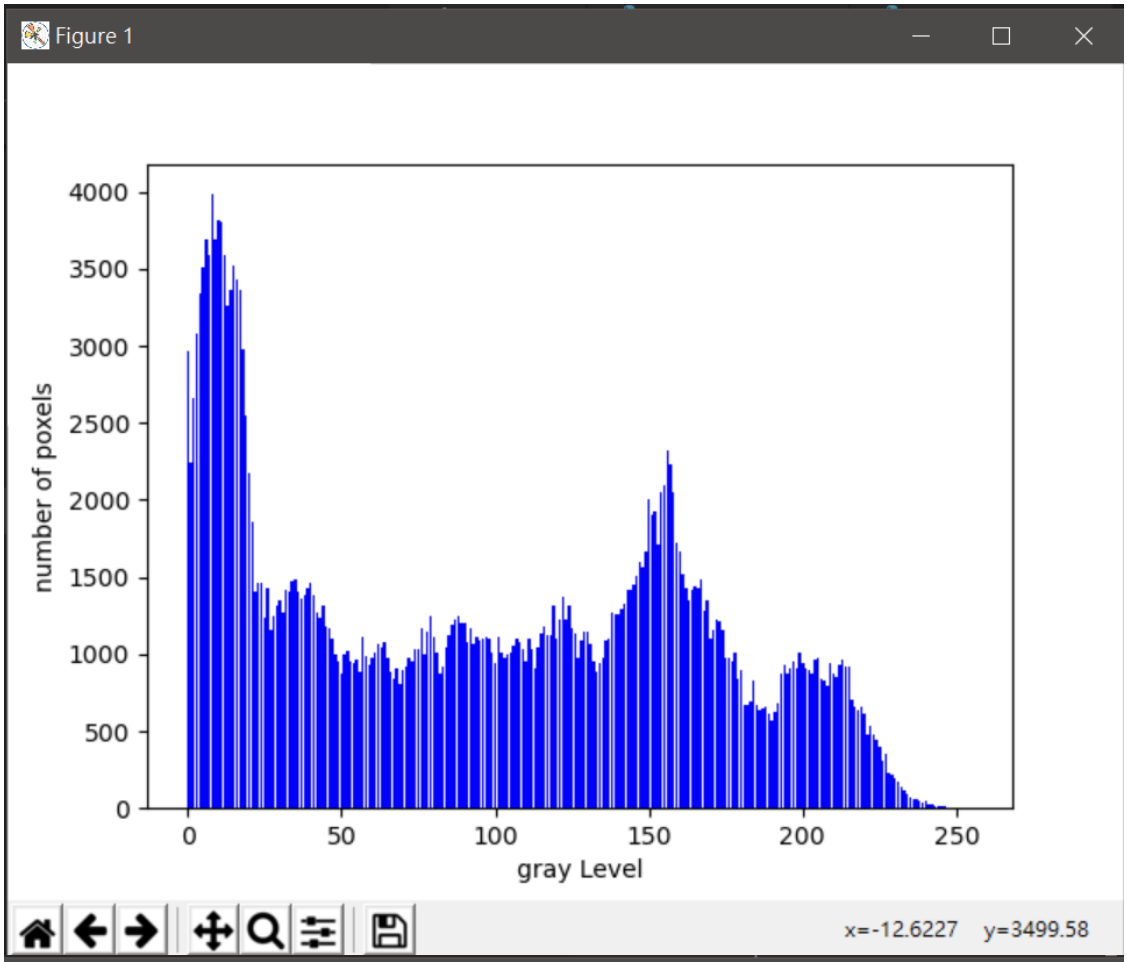


Figure 1
number of poxels
gray Level
x=-12.6227 y=3499.58

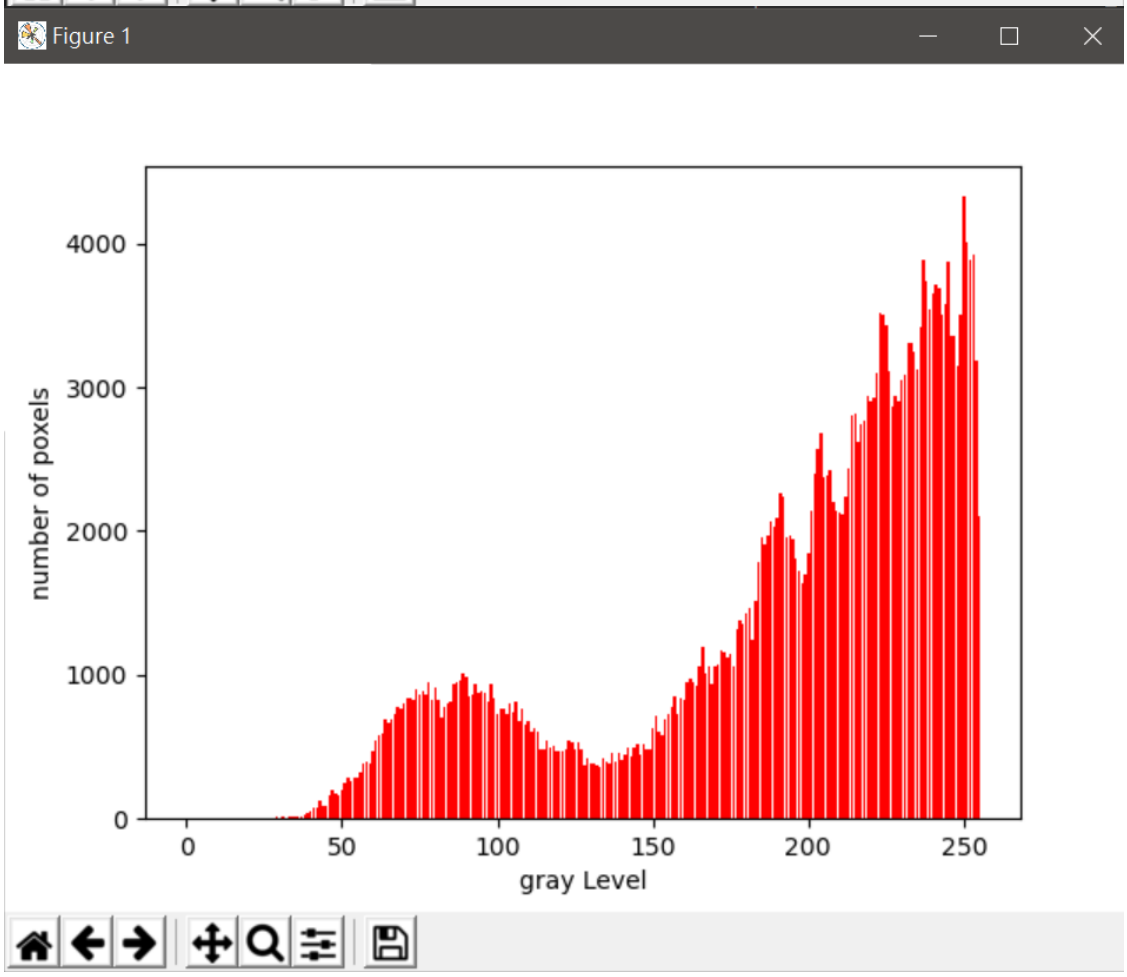
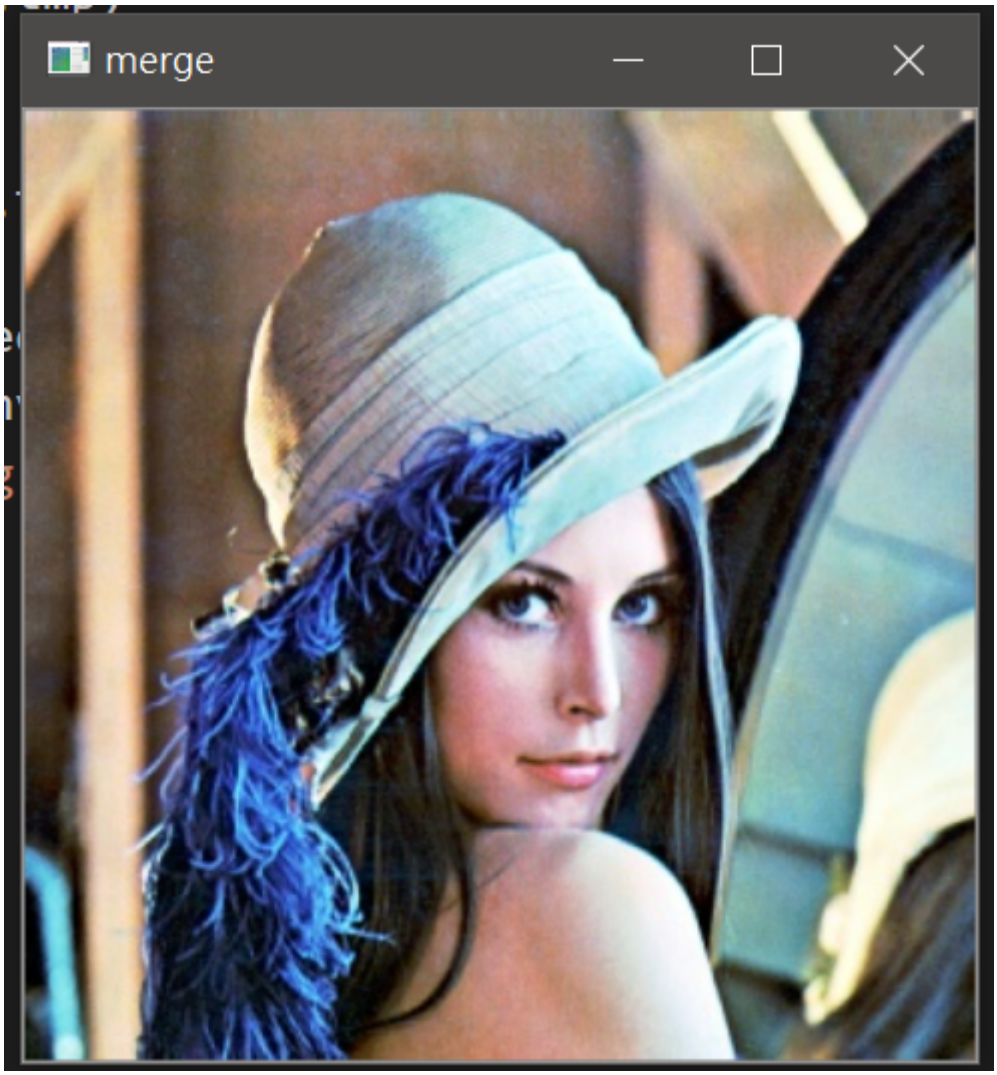


Figure 1
number of poxels
gray Level

- 通道合併



心得

我終於搞懂均化原理了，與機率論有關，這份作業寫了真的很久