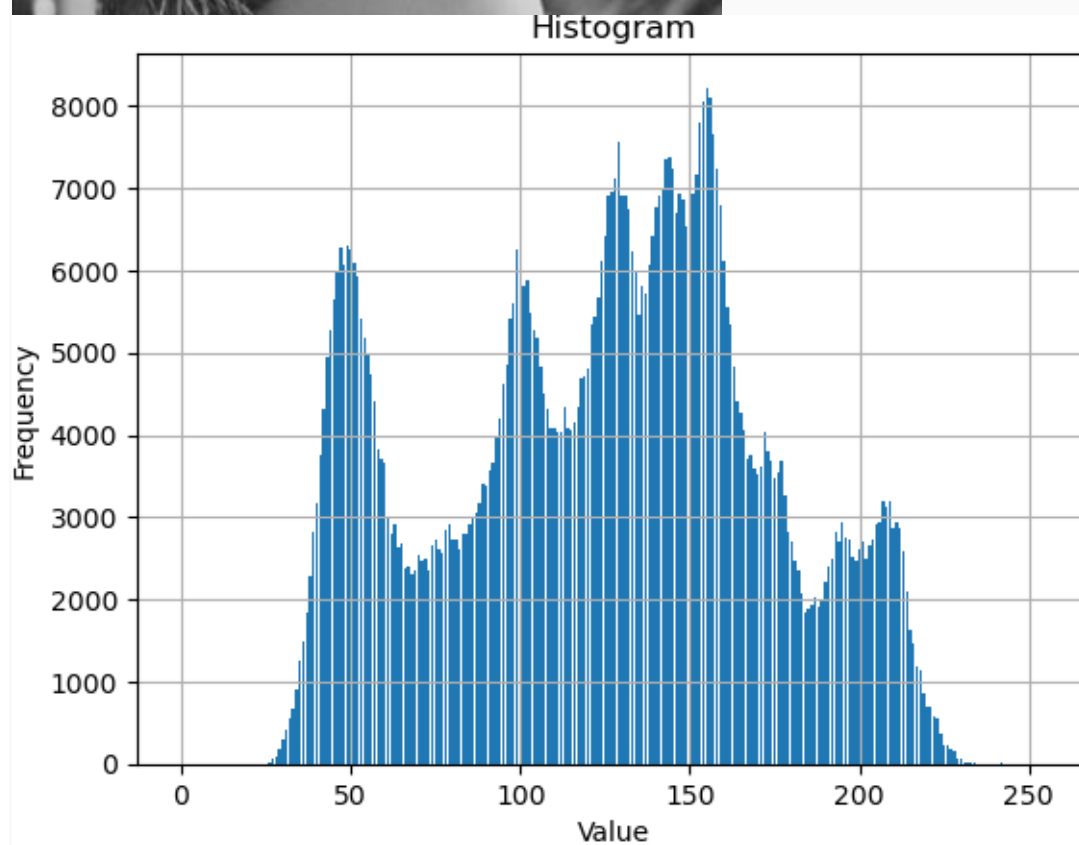


- (a) original image and its histogram

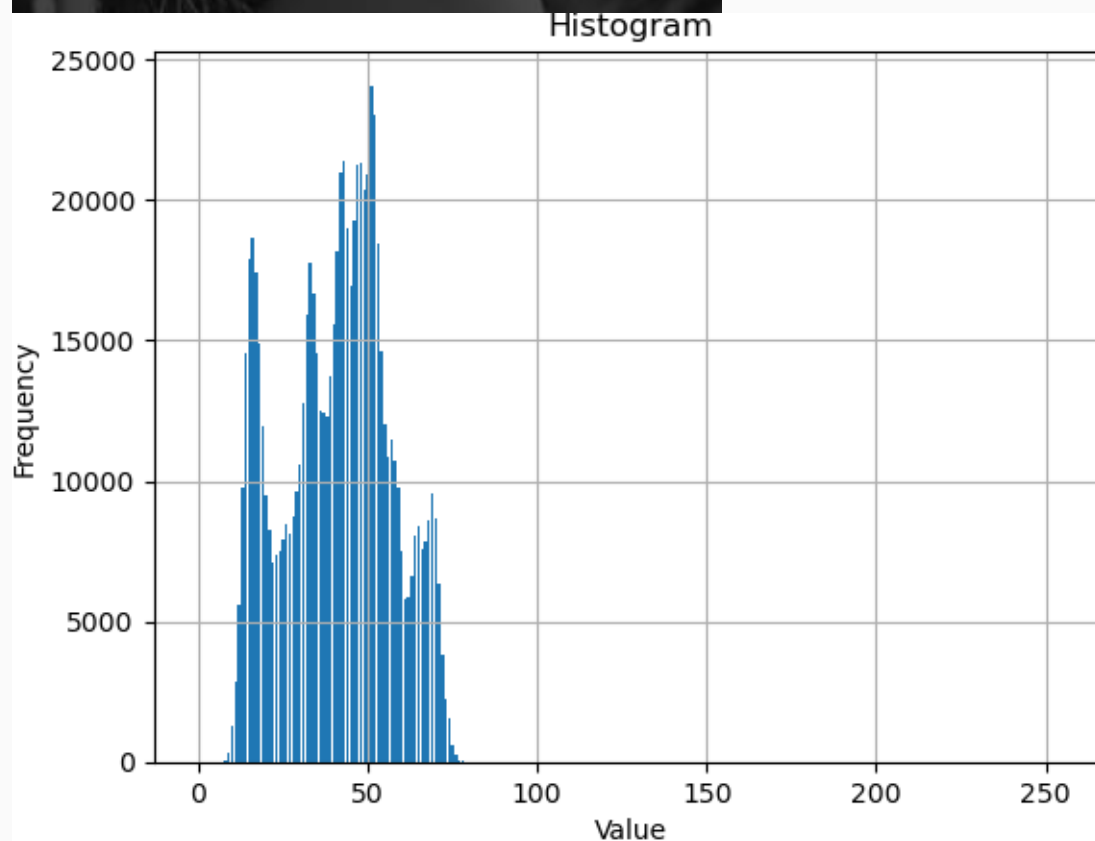
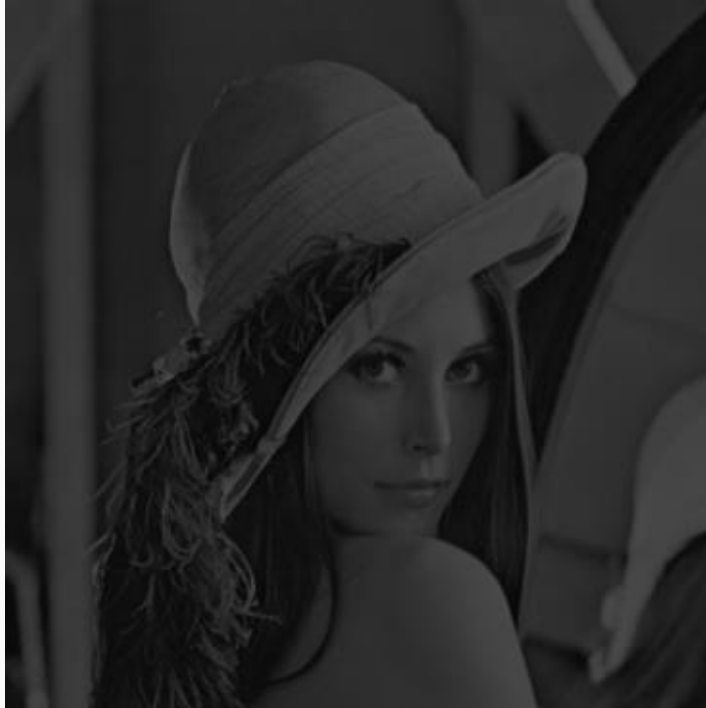


```
pixels = image.flatten()
hist = [0] * 256
for pixel in pixels:
    hist[pixel] += 1
print(hist)
```

```
plt.bar(range(0, 256), hist)
plt.title('Histogram')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.grid(True)
plt.savefig('3a2.png', bbox_inches='tight', pad_inches=0)
plt.show()
```

讀入照片後轉成 array，遍歷每個 pixel 計算各亮度出現次數，統計後用 plt 顯示直方圖。

- (b) image with intensity divided by 3 and its histogram



```
image = cv2.imread('lena.bmp')
image3 = image // 3

plt.imshow(image3)
plt.axis('off')
plt.savefig('3b1.png', bbox_inches='tight', pad_inches=0)
plt.show()
```

將原始照片除以三

```
pixels = image3.flatten()
hist = [0] * 256
for pixel in pixels:
    hist[int(pixel)] += 1

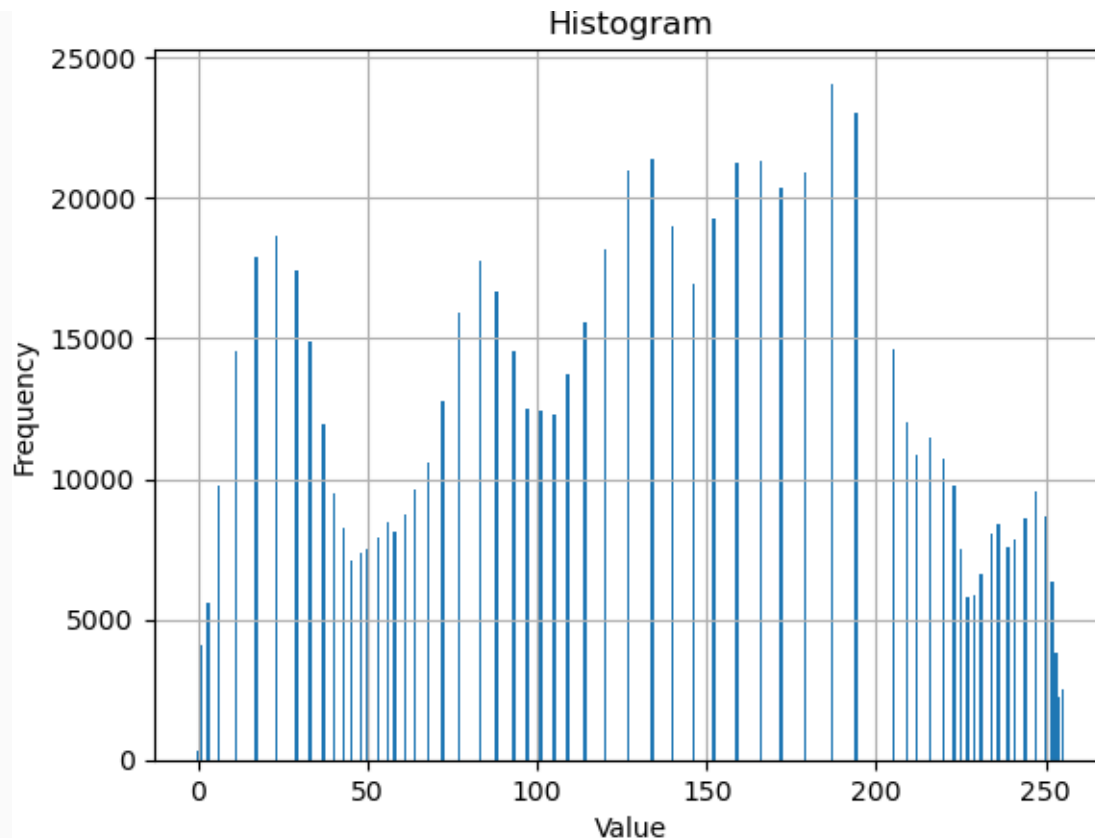
print(hist)

plt.bar(range(0, 256), hist)
plt.title('Histogram')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.grid(True)
plt.savefig('3b2.png', bbox_inches='tight', pad_inches=0)
plt.show()
```

統計各亮度出現次數後轉成直方圖貼上

- (c) image after applying histogram equalization to (b) and its histogram





```

cdf = [0] * 256
cdf[0] = hist[0]
for i in range(1,256):
    cdf[i] = cdf[i - 1] + hist[i]
|
total_pixels = sum(hist)
cdf_normalized = [i * 255 / total_pixels for i in cdf]
cdf_scaled = np.round(cdf_normalized).astype('uint8')
print(cdf_scaled)

```

計算 cdf from (b)

```

equalized_image = np.zeros_like(image3)

for i in range(image3.shape[0]): # 遍歷行
    for j in range(image3.shape[1]): # 遍歷列
        old_intensity = image3[i, j] # 獲取原始像素值
        new_intensity = cdf_scaled[old_intensity] # 映射到新強度值
        equalized_image[i, j] = new_intensity

```

Equalize image from (b)

```

plt.imshow(equalized_image, cmap='gray')
plt.axis('off')
plt.savefig('3c1.png', bbox_inches='tight', pad_inches=0)
plt.show()

```

顯示圖片

```

pixels = equalized_image.flatten()
pixels = [pixel for pixel in pixels]
hist = [0] * 256
for pixel in pixels:
    hist[int(pixel)] += 1

plt.bar(range(0, 256), hist)
plt.title('Histogram')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.grid(True)
plt.savefig('3c2.png', bbox_inches='tight', pad_inches=0)
plt.show()

```

統計計算直方圖