

### (a) original image and its histogram

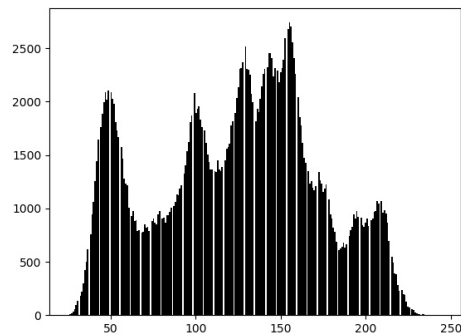
先將照片利用 `ndarray` 讀取

再利用 `flatten` 的方法將該 `ndarray` 變為一維陣列

最後利用 `matplotlib.pyplot` 的 `hist` 方法繪製出直方圖

```
def histogram(img):  
    im = np.array(img)  
    arr = im.flatten()  
  
    plt.hist(arr, bins=256, facecolor='black')  
    plt.savefig("lena_histogram.jpg")  
    plt.show()  
    return 0
```

結果



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

一樣先將照片利用 `ndarray` 讀取

再利用 `flatten` 的方法將該 `ndarray` 變為一維陣列

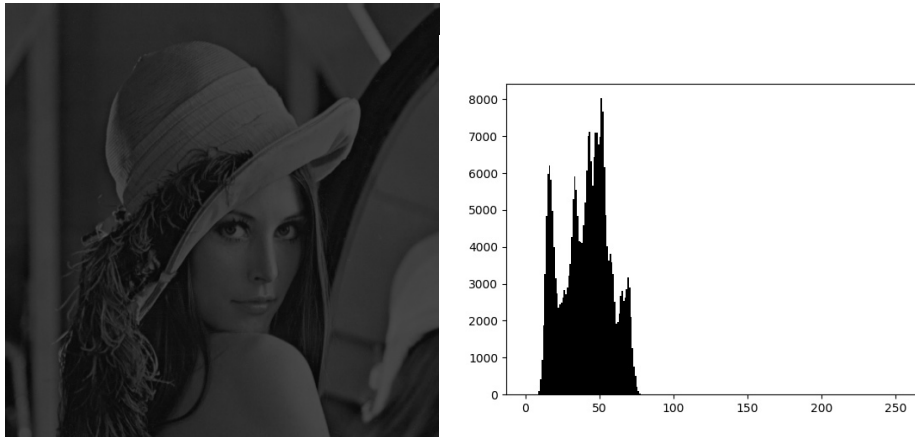
接著直接把整個陣列除以三，得到暗化後的陣列

最後利用 `matplotlib.pyplot` 的 `hist` 方法繪製出直方圖

再把一維陣列轉回 `512*512` 陣列，輸出成圖片

```
def intensity_divided(img):  
    im = np.array(img)  
    arr = im.flatten()  
    arr = arr / 3  
  
    plt.hist(arr, bins=256, range=(0,256), facecolor='black')  
    plt.savefig("lena_intensity_divided.jpg")  
    plt.show()  
  
    arr = arr.reshape([512,512])  
    result = Image.fromarray(np.uint8(arr))  
    result.show()  
  
    return result
```

結果



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

一樣先將照片利用 `ndarray` 讀取

再利用 `flatten` 的方法將該 `ndarray` 變為一維陣列

Step1:

接著創建字典計算每個灰階值出現幾次，並且按順序排列

Step2:

將字典的 `value` 轉換成累積值

Step3:

利用公式將灰階值轉換成均衡化後的值

$$H(v) = (cdf(v) - cdf\_min) / (cdf\_max - cdf\_min) * L$$

把轉換後的值放進陣列，畫出直方圖

```
def histogram_equalization(img):  
    im = np.array(img)  
    arr = im.flatten()  
    count = {}  
    for i in arr: # 利用dict計算每個灰階值出現幾次  
        if i not in count.keys():  
            count[i] = 1  
        else:  
            count[i] += 1  
    count = sorted(count.items())  
  
    cdf = {}  
    x = 0  
    for i in count: # 轉換成累積分佈函數cdf值  
        x += i[1]  
        cdf[i[0]] = x  
    cdf_min = min(cdf.values())  
    cdf_max = max(cdf.values())  
  
    for i, c in cdf.items(): # 均衡化  
        i_equa = (c - cdf_min) / (cdf_max - cdf_min) * 255  
        cdf[i] = int(i_equa)  
    arr_equa = []  
    for i in arr:  
        arr_equa.append(cdf[i])  
    arr_equa = np.array(arr_equa)
```

結果

