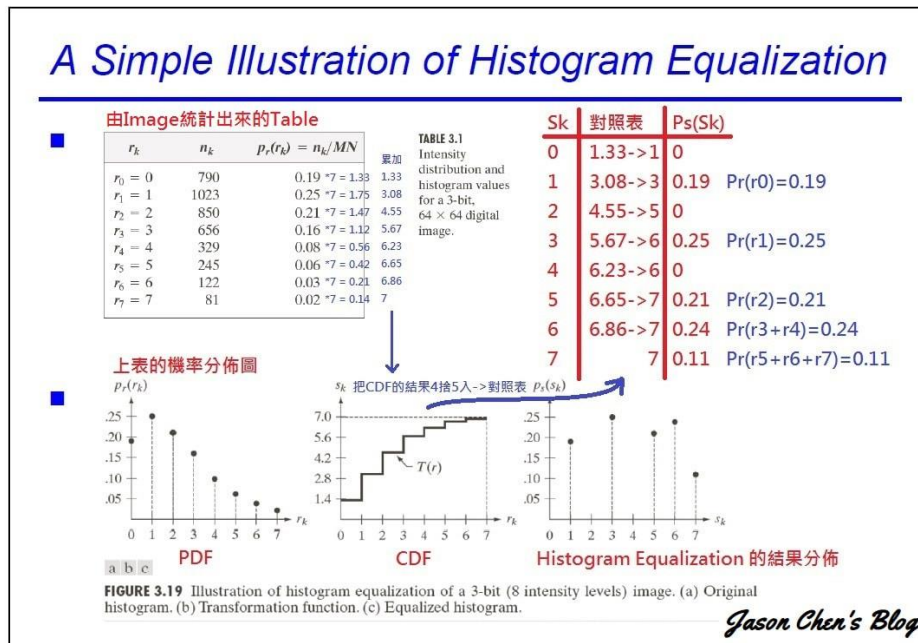


1) Original histogram, equalized histogram, and the processed image after histogram equalization

此題我用兩種方法實作，一個是用 OpenCV 現有的函式，另一個是用 cdf 的方式去 Equalization. 下圖是我參考的網站 ↓。



圖一

Reference: <https://jason-chen-1992.weebly.com/home/-histogram-equalization>

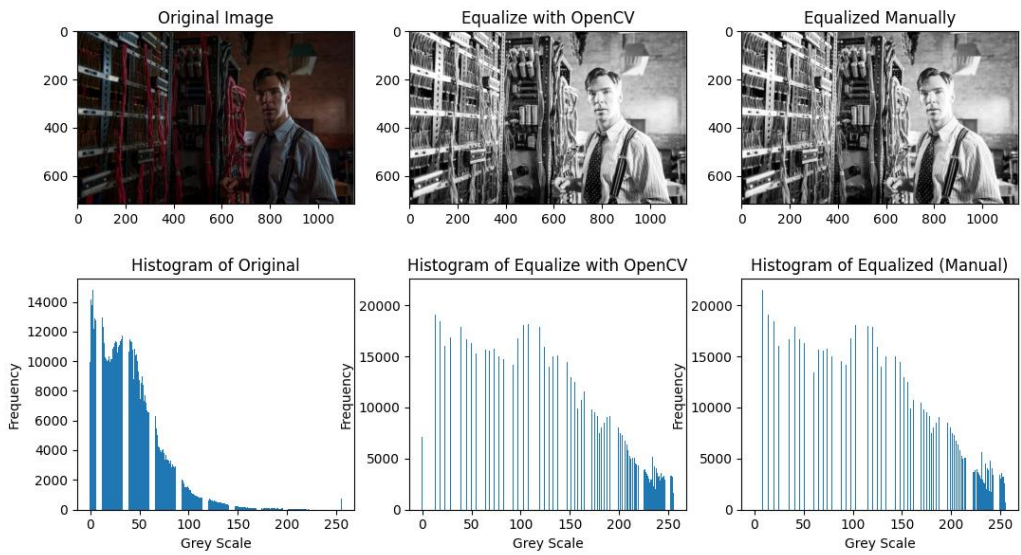
OpenCV:

我用 `cv2.equalizeHist()` 這個 function 實作，再用 `plt.imshow()` 呈現，但因為此函式 default cmap 是 viridis，所以一開始呈現的顏色會很奇怪，只要改成 `cmap = 'grey'` 就會是灰階的形式了。

Manul Version:

用圖一的方法，我自己定義了一個函式叫做 `HistogramEqualization()`，主要流程都是在這個函式做實作。

首先，我先得到這次作業的圖片大小(height, width)，也就是圖一中的 M, N，再利用 M, N 求出圖片像素數(num_pixel)，再利用 `cv2.calcHist()` 得到每個像素出現的頻率，接下來計算 cdf，再將 cdf 除以圖片的總像素數，最後在乘上 255 (因為此為灰階圖)，將得到的值取整數，再將原來的灰階圖像素重新分配，最後一樣再用 `plt.imshow()` 將圖片畫出來。



↑ 結果圖

2) Detected edges in the image after edge detection

利用 Sobel x, Sobel y operator 做邊緣偵測，但在做邊緣偵測之前需要做一些前處理，像是 Gaussian Blur，因此我利用 cv2.GaussianBlur 做了前處理後，再進行對 2D 圖片的 Convolution，先做垂直的邊緣偵測，再做水平的邊緣偵測，兩張圖片的值結合成梯度，得到以下結果↓。

