11/28(日) hw2 D0886261 黃柏儒

直方圖等化

程式：

```python
def Histogram(image,copy):

    rows, cols= image.shape

    size = (rows * cols) # 總 pixel 值
    count = np.zeros(256) # 存放每個像素值出現的次數
    pmf = [] # 存放每個像素值出現的機率

    create = np.zeros((rows, cols, 1), np.uint8) # 新的畫布

    for i in range (size):
        index = copy[i] # 每個像素值設為索引
        count[index] = count[index]+1 # 利用索引存放出現的次數

    for i in range (256):
        pmf.append((count[i]/size)) # 存放出現次數的機率

    cdf = CDF(pmf)  # 傳回所選機率的累加函數

    for x in range(cols):
        for y in range(rows):
            index = image[y,x] # 以向素值為索引
            create[y, x] = round(cdf[index]*255) # 計算轉換函數並返回值給新的畫布

    return create
```

```python
def CDF(pmf): # 傳回所選機率的累加函數
    cdf = []
    for i in range (256):
        sum = 0
        for j in range (0,i+1):
            sum = sum + pmf[j]
        cdf.append(sum)
    return cdf
```

原理：創建陣列存放每個像素值出現的機率，再利用這個機率創建 CDF（累加），利用 CDF，使的取得的座標對應的像素值轉換成對應累加的像素值。

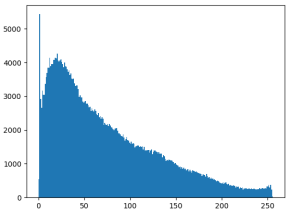
原圖：

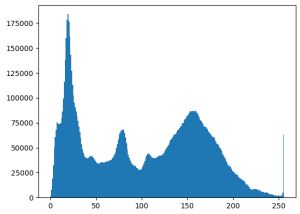
灰階：

處理前：

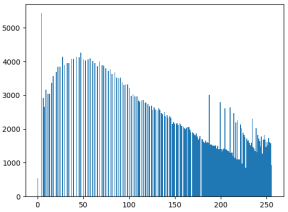
處理後：

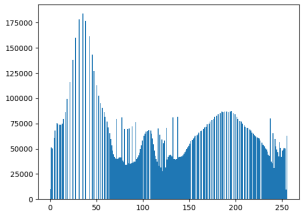
處理前直方圖：



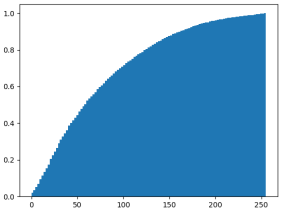


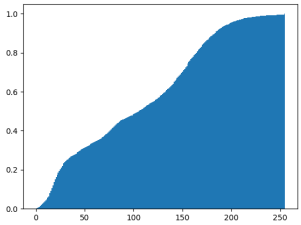
處理後直方圖：



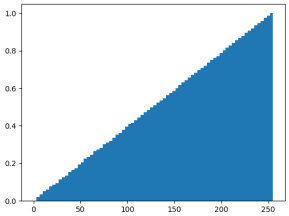


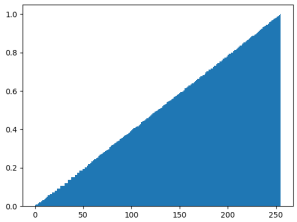
處理前累積直方圖：





處理後累積直方圖：





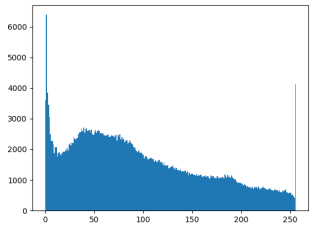
彩色：

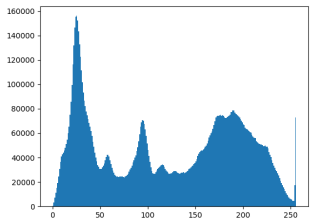
R：

處理前：

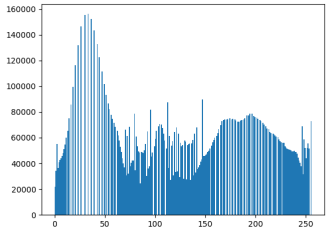
處理後：

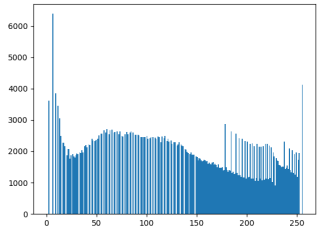
處理前直方圖：



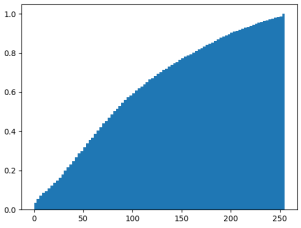


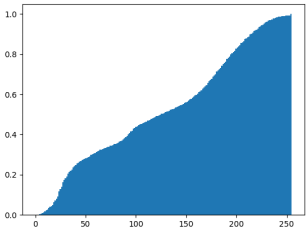
處理後直方圖：



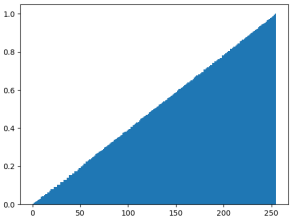


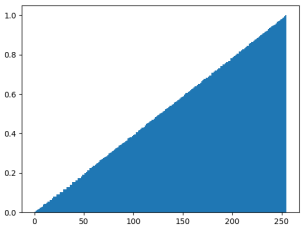
處理前累積直方圖：





處理後累積直方圖：



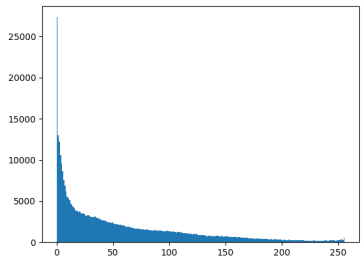


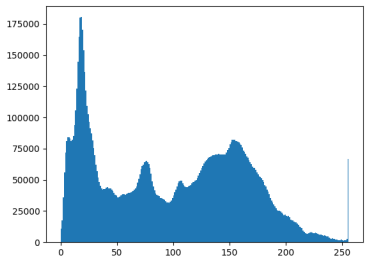
G：

處理前：

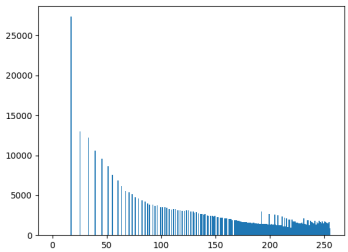
處理後：

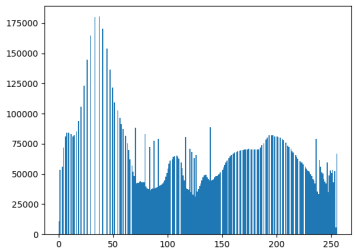
處理前直方圖：



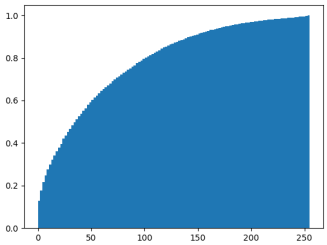


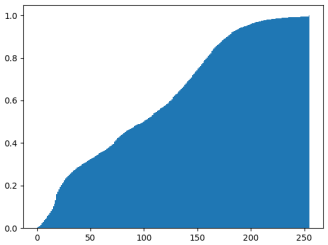
處理後直方圖：



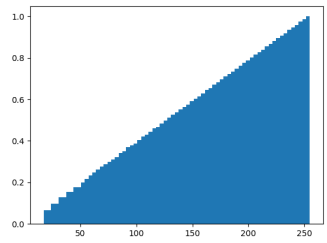


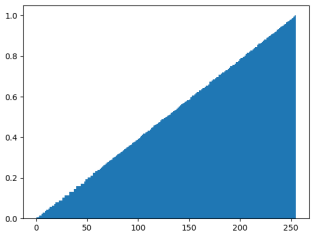
處理前累積直方圖：





處理後累積直方圖：



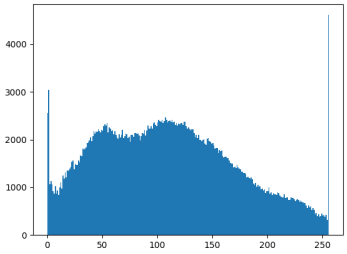


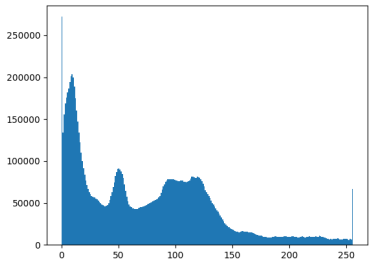
B：

處理前：

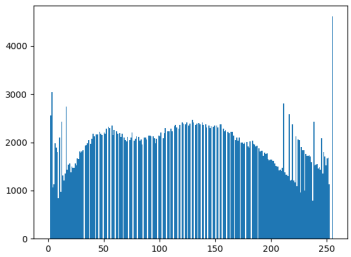
處理後：

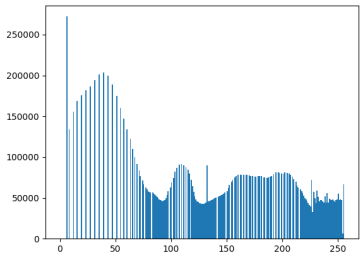
處理前直方圖：



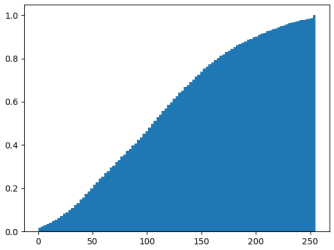


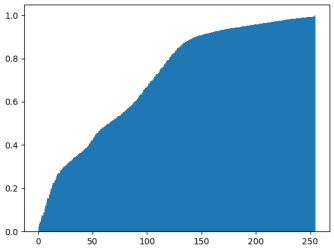
處理後直方圖：



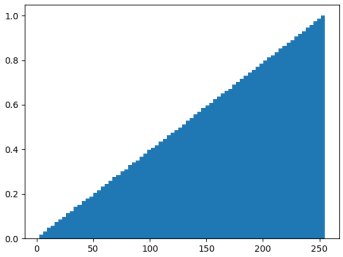


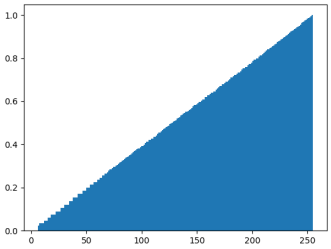
處理前累積直方圖：





處理後累積直方圖：





合併（RGB）：

心得：

雖然最後成功做出來了，但用函式去比對時，雖然大致上直方圖都是一樣的，但還是會有些許的差別，這部分就搞不太懂原因了。