

## Digital Image Processing HW2

M093781 趙宇涵

### I. Function Explanation

以下首先介紹本次作業使用的幾種方法，分別為繪製直方圖、median filter, sharpness, normalization, homomorphic filter, 以及亮度、對比度增強。

#### 1. 繪製直方圖：

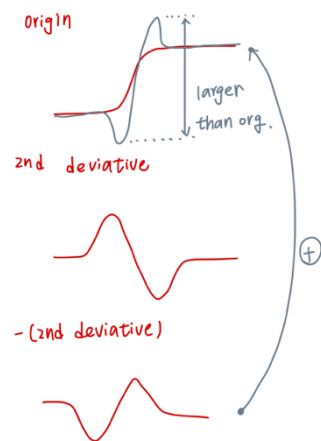
第一個 function 是為了觀察原圖的色彩分佈時使用，首先將影像轉換成灰階分佈，並使用 open cv 內建的功能 calcHist，計算每一個灰度的像素數量。

#### 2. Median filter：

這個功能主要為降低噪點影響（尤其是 salt pepper noise），此除同樣使用 open cv 內建功能 medianBlur。其原理為統計鄰近像素的灰階值，並選用中間值作為該像素點的值，因此可以有效解決像是黑白噪點這種極端值的問題。此處我設計可以自選 kernel 的大小，kernel 越大，輸出影像會越模糊。

#### 3. Sharpness:

課程中銳利化的部分，提到可以使用二階微分-Laplacian filter 加上原圖的方式，加強邊緣劇烈變化的地方，達到銳利化的目標。



圖一、-Laplacian filter 加上原圖示意圖

#### 4. Normalization:

此功能的目標是將灰階分佈集中的影像拉開，此處先將色彩圖拆分 RGB 三個圖層，接著使用 opencv 的 normalization，將最大值與最小值拉至 255 與 0，其他值與之呈現性變化，公式如下：

$$dst(i,j) = \frac{[src(i,j) - \min(src(x,y))] * (max - min)}{\max(src(x,y)) - \min(src(x,y))} + \min$$

##### 5. 亮度、對比度：

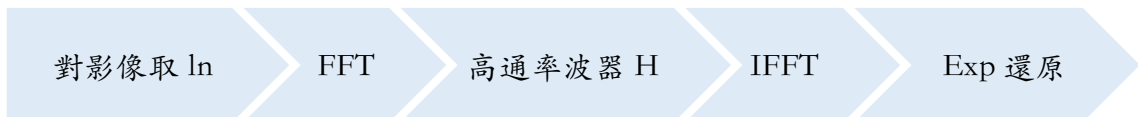
增加對比度的主要目標，就是讓黑的更黑、白的更白。因此此處參考使用三角函數的方法，將影像做亮度與對比度的調整。此處有兩個變數：亮度與對比，將兩個值除以 255 作為係數，並使用以下公式線性調整：

$$img = (img - 127.5 * (1 - B)) * k + 127.5 * (1 + B)$$

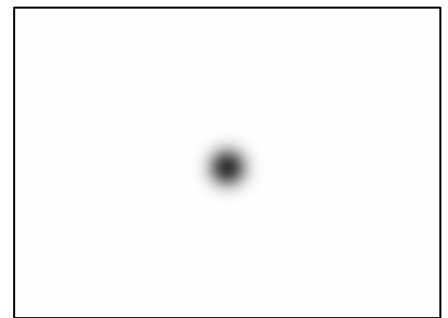
其中B 為亮度係數，係數越大，輸出亮度越亮；k 為對比度係數，k 值越大對比度越大，此處也可以設定為負值，減少對比度。

##### 6. Homomorphic filter：

這個功能主要是強化影像中的細節，做法是把影像從空間域轉換為頻域，並過使用高通濾波抑制低頻，因此在暗處的細節就會變得更明顯，此方法的流程圖如下：



此處的高通濾波器經過高斯模糊處理，臨界邊緣較為平滑，視覺化後如右圖所示，此高通濾波器的變數為欲過濾的頻率 threshold，也就是圓的半徑，圓的半徑越大，能通過的資訊越少，輸出影像的色彩較暗，僅有高頻的影像可以顯示，如下圖三所示，另外一個變數為高斯模糊的 kernel 大小。





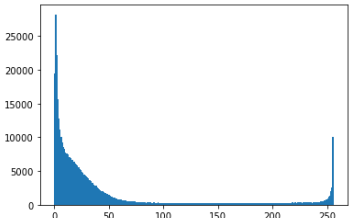
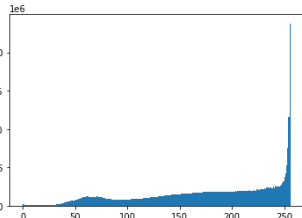
圖二、高通濾波示意圖



圖三、不當 threshold 結果，僅有高頻區域被保留



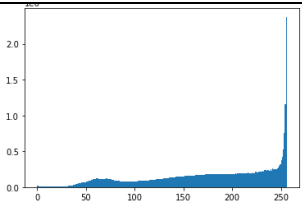
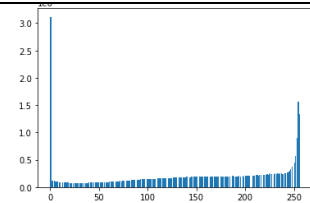
## II. 影像分析與結果

### 1. Input1.bmp

input	output
	
	

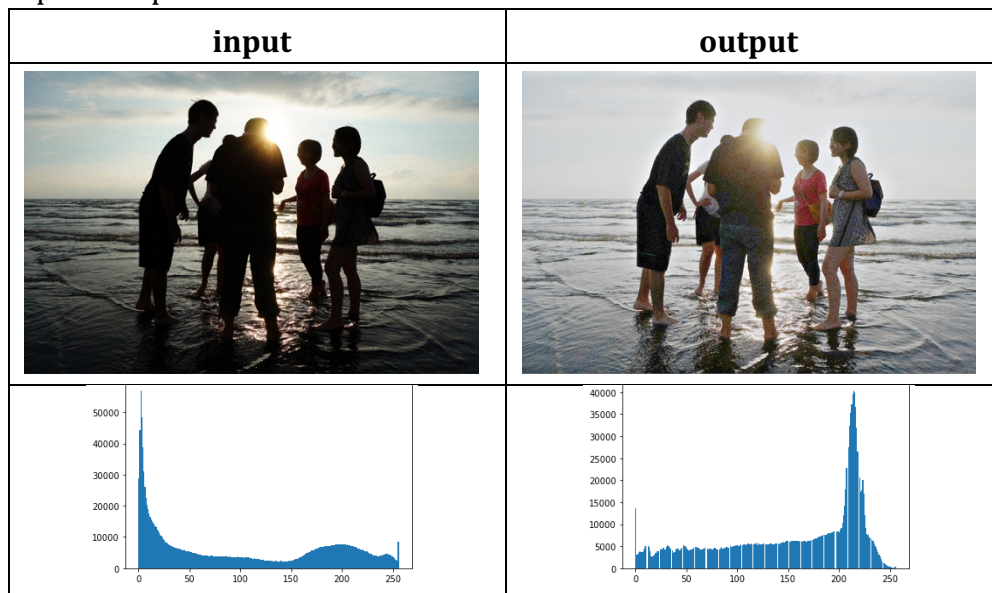
說明：input1 分佈有許多暗處，目標是把陰影處的細節加強，因此使用 homomorphic filter, contrast 兩種方法。

### 2. Input2.bmp

input	output
	
	

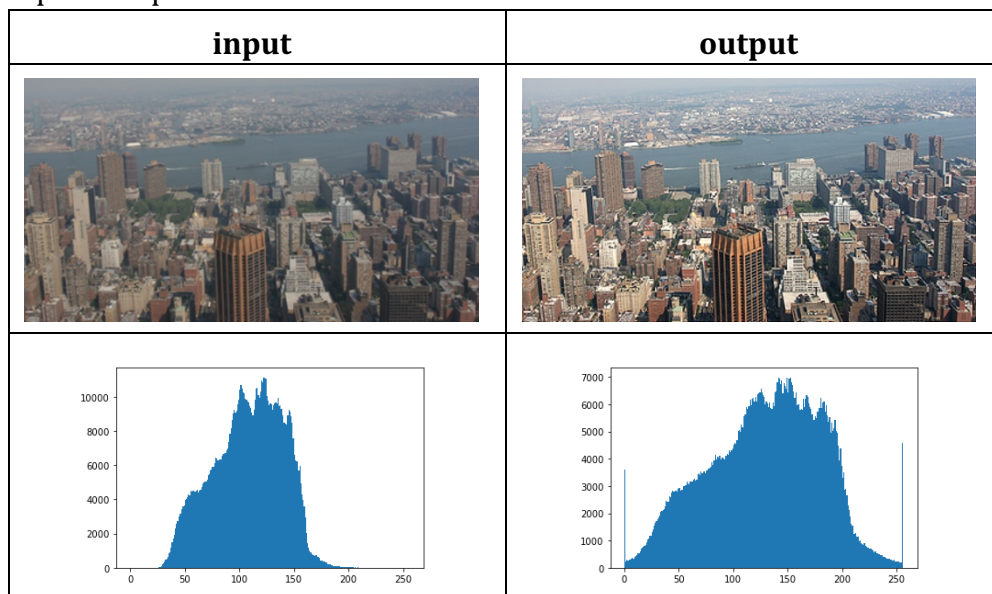
說明：input2.bmp 原圖的問題是色彩過白，因此我使用 contrast 增加對比度，並使用 blur 去除一些雜訊，sharpness 加強銳度。

### 3. Input3.bmp



說明：input3.bmp 的問題是背光嚴重，人的細節完全看不清楚，因此使用 homomorphic filter 加強人物細節，最後再加強對比度。

### 4. Input4.bmp



說明：input4.bmp 的問題是輪廓模糊，且色彩不鮮豔，因此我使用 contrast 增加亮度與對比度、sharpness 加強邊緣銳度。

## III. Reference

1. [https://www.wongwonggoods.com/python/python\\_opencv/opencv-modify-contrast/](https://www.wongwonggoods.com/python/python_opencv/opencv-modify-contrast/)
2. <https://stackoverflow.com/questions/64284739/homomorphic-filtering-on-the-frequency-domain-python-opencv>