**Digital Image Processing HW2**

M093781 趙宇涵

1. Function Explanation

以下首先介紹本次作業使的幾種方法，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。

1. 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。

1. 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

1. 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：

一張含有 文字 的圖片

自動產生的描述

1. 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：

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。

1. 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：

Exp還原

IFFT

高通率波器H

FFT

對影像取ln

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

一張含有 夜晚 的圖片

自動產生的描述

圖三、不當threshold結果，僅有高頻區域被保留

1. 影像分析與結果
2. Input1.bmp

|  |  |
| --- | --- |
| **input** | **output** |
| 一張含有 室外, 城市 的圖片  自動產生的描述 | 一張含有 舊, 髒 的圖片  自動產生的描述 |
|  |  |

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。

1. Input2.bmp

|  |  |
| --- | --- |
| **input** | **output** |
| 一張含有 室外, 樹, 大自然, 泥土 的圖片  自動產生的描述 | 一張含有 樹, 室外, 大自然, 水 的圖片  自動產生的描述 |
|  |  |

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。

1. Input3.bmp

|  |  |
| --- | --- |
| **input** | **output** |
| 一張含有 室外, 天空, 水, 海灘 的圖片  自動產生的描述 | 一張含有 水, 室外, 個人, 人 的圖片  自動產生的描述 |
|  |  |

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。

1. Input4.bmp

|  |  |
| --- | --- |
| **input** | **output** |
| 一張含有 山, 城市, 室外 的圖片  自動產生的描述 | 一張含有 城市, 山, 室外, 人群 的圖片  自動產生的描述 |
|  |  |

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。

1. Reference
2. ﻿<https://www.wongwonggoods.com/python/python_opencv/opencv-modify-contrast/>
3. ﻿<https://stackoverflow.com/questions/64284739/homomorphic-filtering-on-the-frequency-domain-python-opencv>