

影像處理概論 HW2

題目：Image Sharpening

409410095 王子奕

繳交期限：May 9, 2023

繳交日期：May 8, 2023

- Technical description:

(兩張test images: “blurry_moon.tif” 、

“skeleton_orig.bmp” 需與 hw2.py 存在同個資料夾下，此作業使用了 python 的 OpenCV 和 numpy 套件)。

1. 首先，將兩張test images分別讀入兩變數img1、img2。(如下圖)

```
import cv2
import numpy as np

# 讀取影像
img1 = cv2.imread("blurry_moon.tif", cv2.IMREAD_GRAYSCALE)
img2 = cv2.imread("skeleton_orig.bmp", cv2.IMREAD_GRAYSCALE)
```

2. 定義Laplacian operator, 並且將做過Laplacian的images存入變數'sharpened_img1'和'sharpened_img2'(如下圖), 變

數'laplacian_kernel'為, Laplacian operator可以將一個像素的鄰域中的亮度值與該像素自身的亮度值進行差分，以檢測出邊緣和細節的變化。在Laplacian operator中，通常使用以下的3x3矩陣來對像素進行差分計算：

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 1 & -4 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

```
# 定義Laplacian operator
laplacian_kernel = np.array([[0, 1, 0], [1, -4, 1], [0, 1, 0]])
sharpened_img1 = cv2.filter2D(img1, -1, laplacian_kernel)
sharpened_img2 = cv2.filter2D(img2, -1, laplacian_kernel)
```

3. 定義High-boost filtering, 先對原圖做Gaussian blur後並進行加權疊加，將處理後的images存入變數'sharpened_img1_2'和'sharpened_img2_2'(如下圖)

```
# 定義High-boost filtering
k = 1.5 # k is high-boost filter parameter
blurred_img1 = cv2.GaussianBlur(img1, (3, 3), 0) # Gaussian blur
sharpened_img1_2 = cv2.addWeighted(img1, k, blurred_img1, -k+1, 0)
blurred_img2 = cv2.GaussianBlur(img2, (3, 3), 0) # Gaussian blur
sharpened_img2_2 = cv2.addWeighted(img2, k, blurred_img2, -k+1, 0)
```

4. 將兩test images原圖分別顯示於個別輸出視窗內。(如下圖)

```
# 顯示原影像
cv2.imshow("Original Image 1", img1)
cv2.imshow("Original Image 2", img2)
```

5. 輸出所有processed images

```
# 顯示處理後的影像
cv2.imshow("Sharpened Image 1 (Laplacian)", sharpened_img1)
cv2.imshow("Sharpened Image 2 (Laplacian)", sharpened_img2)
cv2.imshow("Sharpened Image 1 (High-boost)", sharpened_img1_2)
cv2.imshow("Sharpened Image 2 (High-boost)", sharpened_img2_2)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

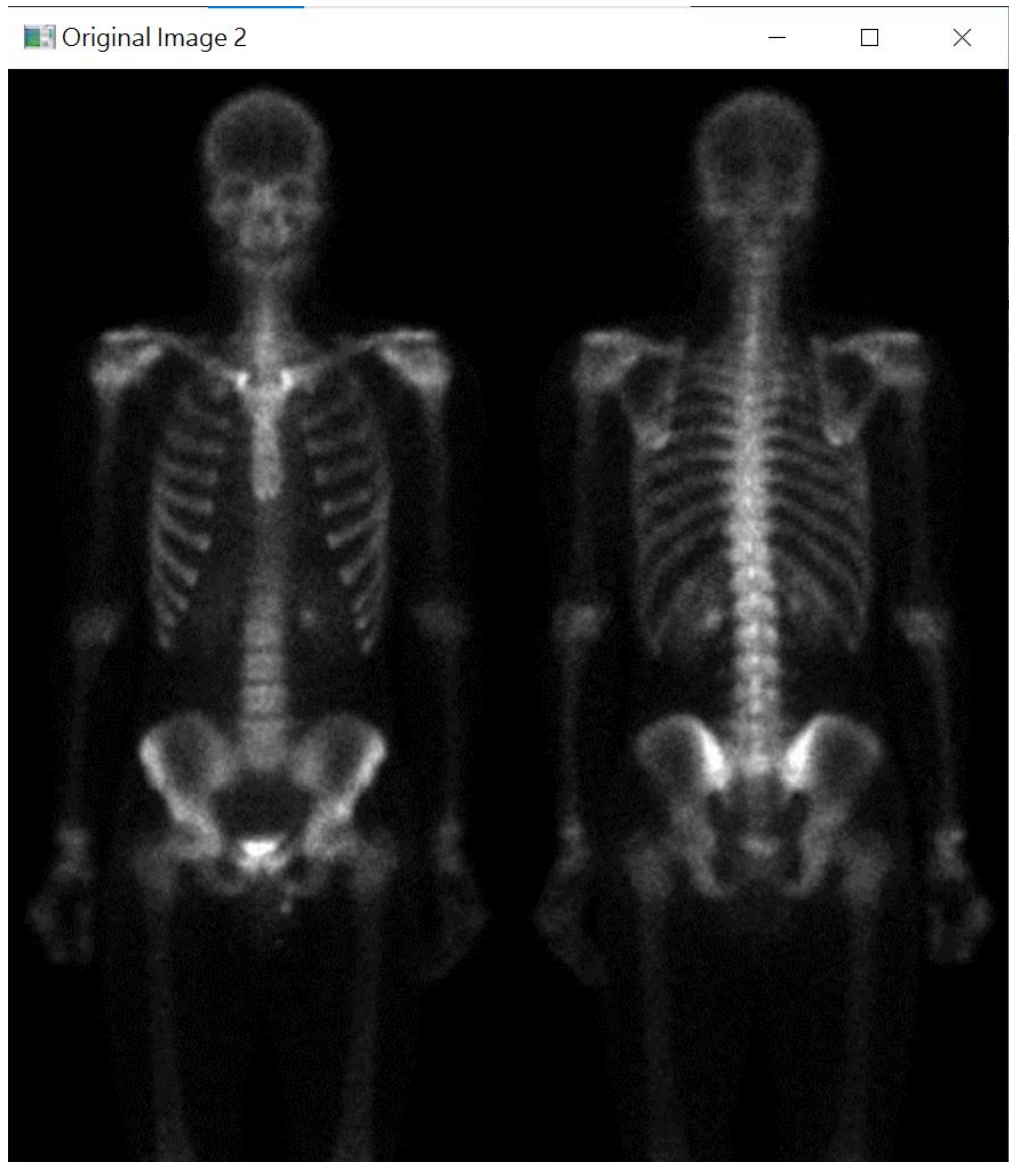
- Experimental results:

1. 該程式的執行結果總共為六個輸出視窗，Original Image 1

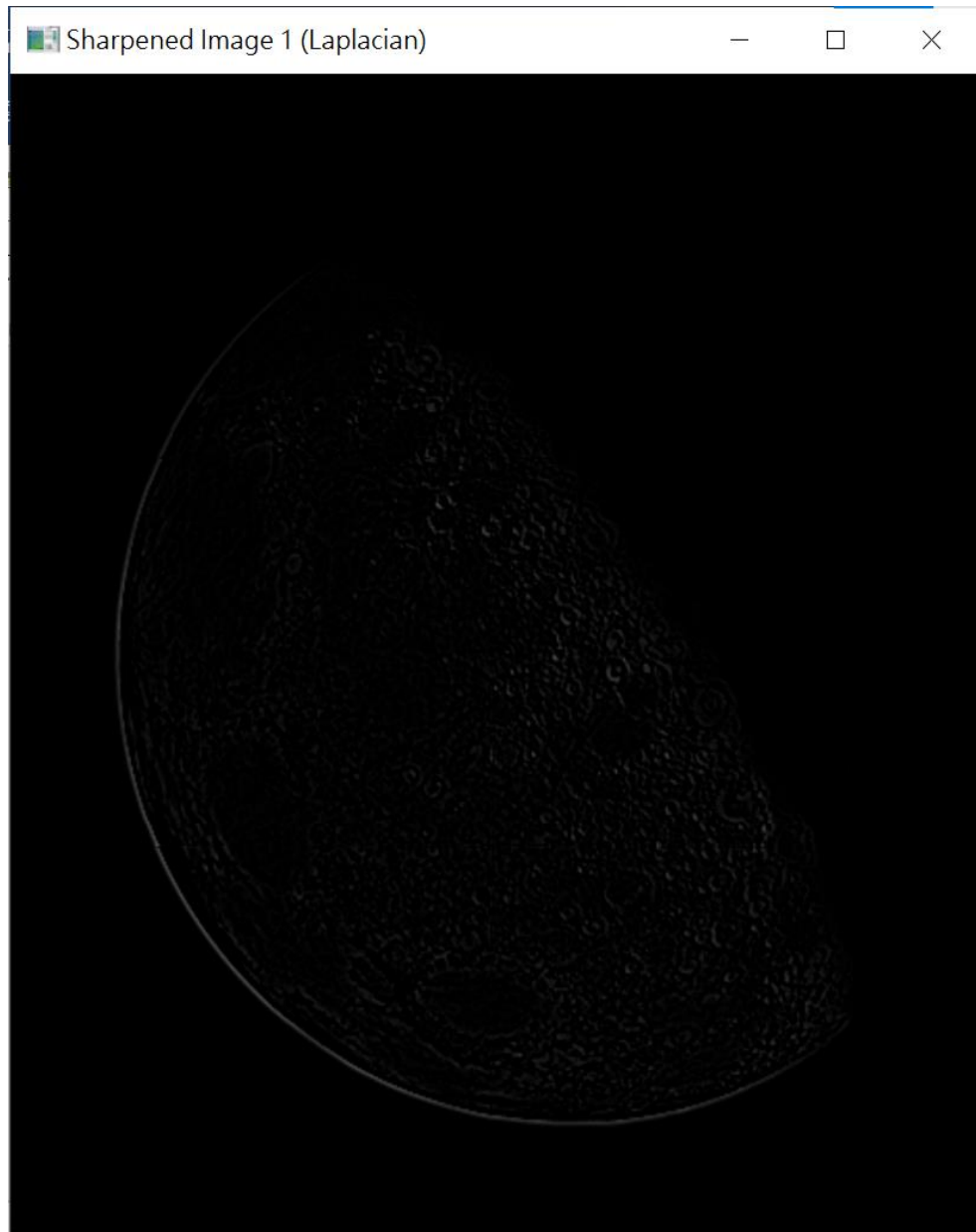
視窗存放blurry_moon.tif原圖。(如下圖)



2. Original Image 2視窗存放skeleton_orig.bmp原圖。(如下圖)

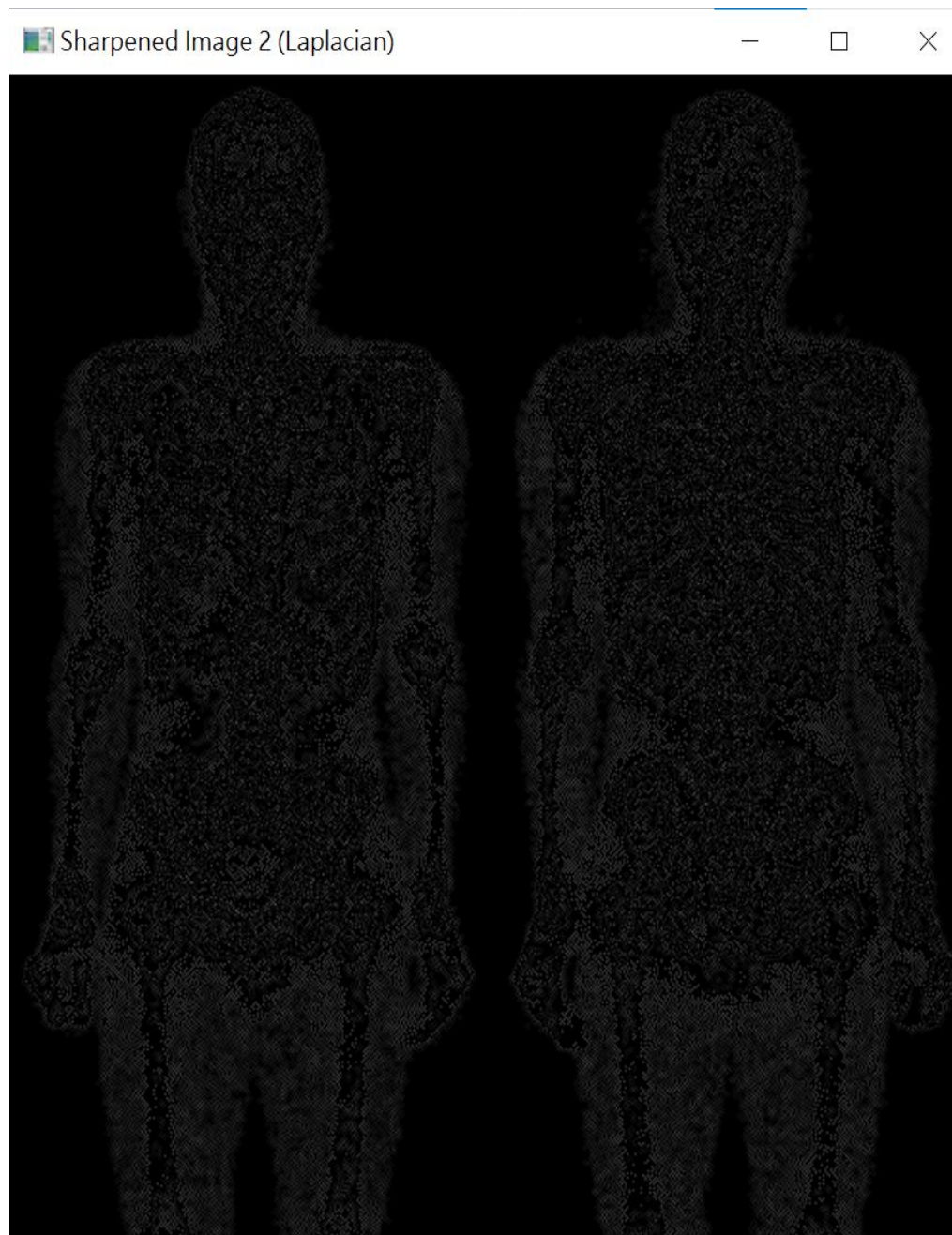


3. Sharpened Image 1 (Laplacian)視窗存放blurry_moon.tif
經過Laplacian處理後的圖。(如下圖)



4. Sharpened Image 2 (Laplacian)視窗存放

skeleton_orig.bmp 經過Laplacian處理後的圖。(如下圖)



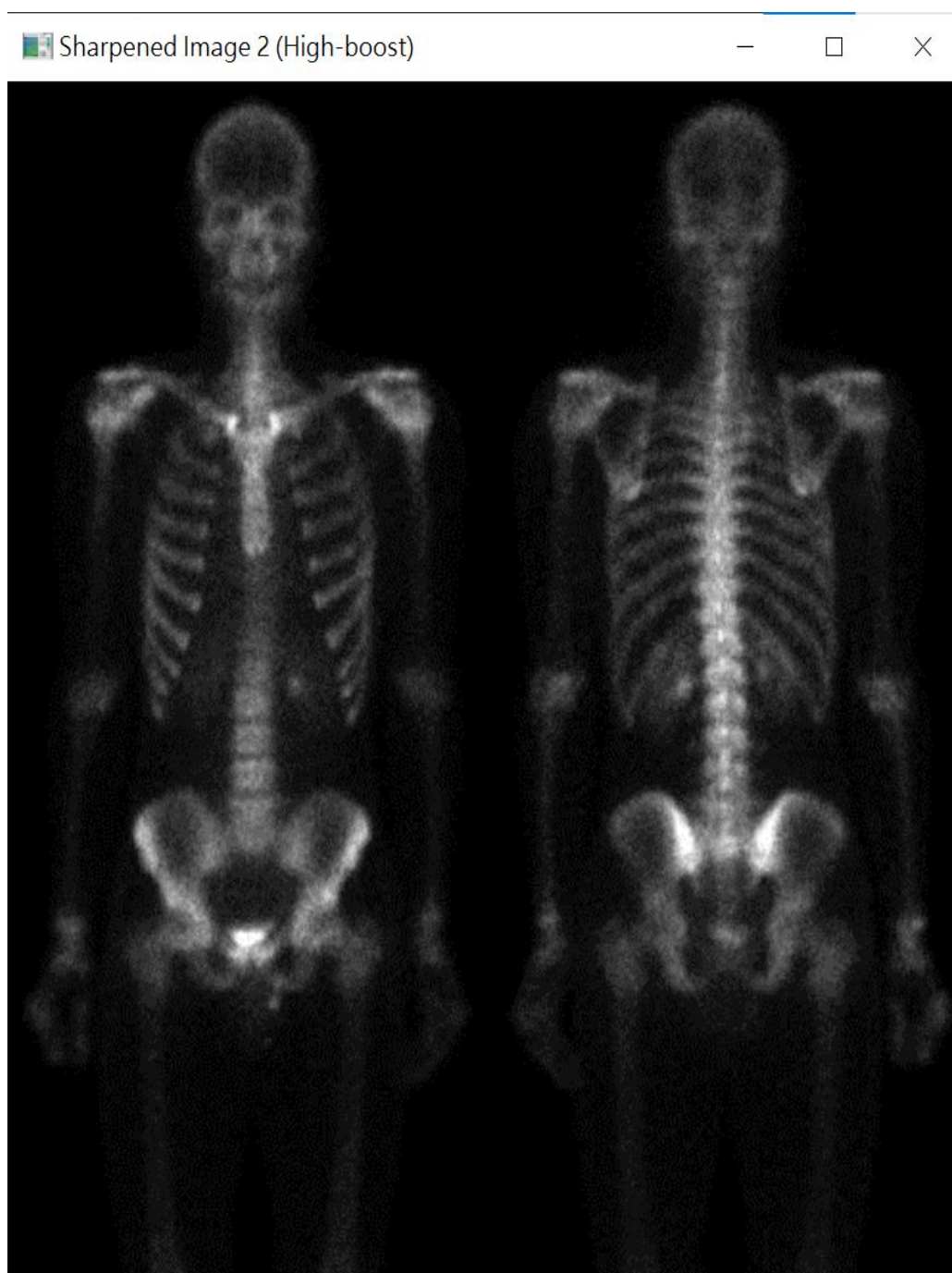
5. Sharpened Image 1 (High-boost)視窗存放blurry_moon.tif

經過High-boost處理後的圖。(如下圖)



6. Sharpened Image 2 (High-boost)視窗存放

skeleton_orig.bmp經過High-boost處理後的圖。(如下圖)



● Discussions:

Image Sharpening(影像銳化):通過增強影像的邊緣和細節，可以提高影像的清晰度和對比度。拉普拉斯算子是一種二階微分算子，可以用於檢測影像中的邊緣和細節。拉普拉斯算子(Laplacian operator)可以將像素值變化較大的區域標記為邊緣，將像素值變化較小的區域標記為平滑區域。因此，拉普拉斯算子可以用於對影像進行邊緣檢測和影像銳化。拉普拉斯影像顯示了影像中每個像素的變化情況，其中邊緣和細節顯示為高亮區域，平滑區域顯示為暗區域。拉普拉斯影像可以通過增加其亮度和對比度，得到一張銳化影像。具體來說，可以將原始影像和拉普拉斯影像進行加權疊加，得到一張銳化影像。高提升濾波器是一種通過增強影像的高頻部分，從而提高影像的對比度和清晰度的方法。具體來說，高提升濾波器可以通過對原始影像進行平滑操作，得到一張平滑影像，然後將原始影像和平滑影像進行加權疊加，得到一張Sharpened Image。高提升濾波器的主要參數是平滑影像和原始影像的權重比例，通過調整這個比例可以控制銳化的程度。當權重比例越大時，銳化的程度越高, 透過這次homework讓我更能夠了解兩者的實際輸出結果和運作方法。

● References and Appendix:

--references:

1. <https://towardsdatascience.com/image-processing-with-python-blurring-and-sharpening-for-beginners-3bcebec0583a>
2. <https://www.geeksforgeeks.org/image-sharpening-using-laplacian-filter-and-high-boost-filtering-in-matlab/>
3. <https://towardsdatascience.com/image-processing-with-python-blurring-and-sharpening-for-beginners-3bcebec0583a>