

HW4 - Sharpening

111550129 林彥亨

I. Method

本作業主要實作兩種影像銳化方法：spatial domain 與 frequency domain 的 Laplacian filter，並應用於 moon.jpg 上。

1. Laplacian filter in spatial domain

我們使用了兩種 Laplacian kernel (4-neighbor 與 8-neighbor) 來計算二階導數。

0	1	0
1	-4	1
0	1	0

4-neighbor

1	1	1
1	-8	1
1	1	1

8-neighbor

在 2D convolution 的實作中，為了避免邊界的遺失，對原始圖片先做 zero padding，並且與 kernel 做元素乘積後加總，得到對應位置的卷積結果。

Sharpening 的部分，我們將原圖去減掉 Laplacian 結果後得到 sharpen 的圖像。以下是我們利用的公式：

$$g(x,y) = f(x,y) - c \cdot \nabla^2 f(x,y)$$

2. Laplacian filter in frequency domain

使用 numpy 的 np.fft.fft2() 將影像轉為頻率域表示 $F(u, v)$ ，並使用 np.fft.fftshift() 將頻譜中心對齊至影像中心。

建立 Laplacian filter，利用公式：

$$H(u, v) = \left(-4\pi^2 \left[\left(u - \frac{P}{2} \right)^2 + \left(v - \frac{Q}{2} \right)^2 \right] \right)$$

將 Laplacian filter 代回頻譜上：




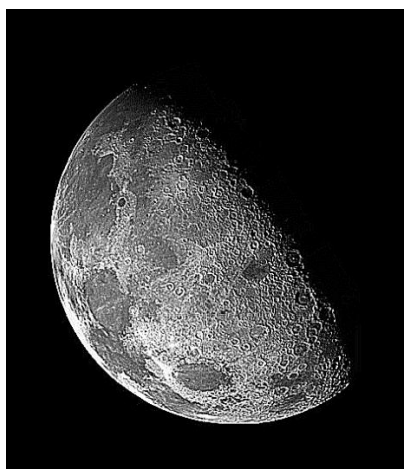
$$G(u, v) = H(u, v) \cdot F(u, v)$$

再透過 np.fft.ifftshift() 和 np.fft.ifft2() 還原回空間域，並取實數部分得到 Laplacian 結果。

為了讓 Laplacian 結果數值適用於後續運算，我們將其縮放至 $[-1, 1]$ ，再進行銳化。

II. Result

1. Result images

Original image	Spatial domain – Kernel_1
	
Frequency domain	Spatial domain – Kernel_2
	

2. Comparison

Spatial Domain	Frequency Domain
<ul style="list-style-type: none">銳化後邊緣變得明顯，細節更清晰，特別是月球坑洞與表面紋理。使用 8-neighbor kernel 強化範圍較大，整體銳利度提升顯著。算法簡單、執行快速，適合即時處理與硬體實作	<ul style="list-style-type: none">銳化效果較空間域平滑自然，邊緣過渡柔和，但仍保留細節強化效果。頻率濾波器設計為全域性。Laplacian 在頻率域上能更精準地操作不同頻率區段，但計算成本較高。



III. Feedback

透過本次實作，我更深入理解了空間域與頻率域 Laplacian 濾波在影像銳化上的應用。空間域方法直觀且實作簡單，但在邊界處理上容易出現不自然的黑邊，需注意 padding 的選擇；頻率域方法雖然實作較為繁瑣，卻能提供更平滑的邊緣過渡與全域性的增強效果，適合需要高品質輸出的場景。這次練習不僅加深我對影像處理核心概念的理解，也讓我學會根據實際需求選擇適合的處理方式。