

## 影像處理與電腦視覺 作業三

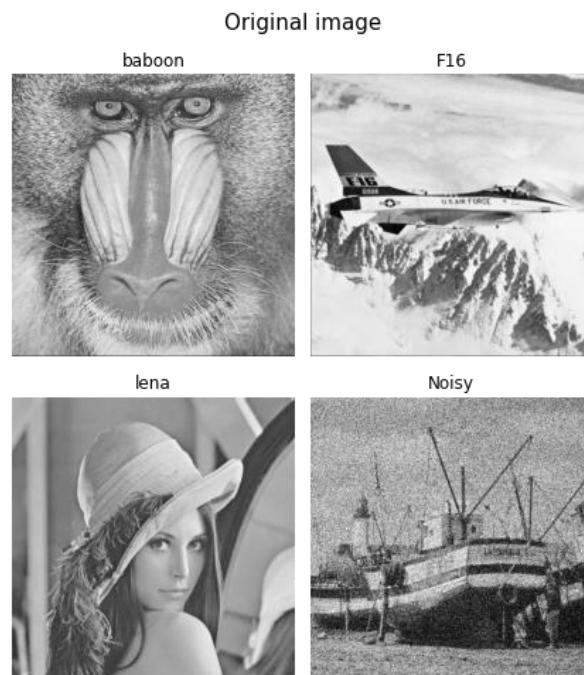
統計所 李權恩 R26094022

### Introduction

在影像處理有許多不同的任務，有更加強調圖片細部結構的邊緣銳化 (Edge enhancement)，利用不同平滑核濾波器 (Smooth kernel filter) 對圖片的像素值進行卷積 (Convolution) 運算來達到降噪 (De-noising) 目的，這之中又為了改進不同的缺點發展出許多不同特色的濾波器 (Filter)，此外還有許多不同目標的任務。

影像除了考慮在空間域 (Spatial domain) 之外，也可以視為一種多維度的訊號，將其轉換到頻率域 (Frequency domain) 上，除了更具物理含意外，透過轉換的數學性質，原本在空間域上的卷積計算再轉換到頻率域之後更為簡易，本次作業主要考慮了影像處理在頻率域上的方法，轉換的方式有離散傅立葉轉換 (Discrete Fourier Transform; DFT) 及離散餘弦轉換 (Discrete Cosine Transform; DCT)，也探討了關於在頻率域上的濾波器包含：低通濾波器 (Low-pass filter)、高通濾波器 (High-pass filter) 及其他不同特色的方法。

本次主要使用的測試圖片如下：



最後，本次作業主要以 Python 完成，相關程式碼於此連結中：

<https://github.com/Quan-En/DIPCV/tree/main/assignment3>

相關函數如何呼叫與使用於此：

<https://github.com/Quan-En/DIPCV/blob/main/assignment3/hw3.ipynb>

## Methodology

### 1. Discrete Fourier Transform (DFT)

DFT 主要是將空間域上的訊號轉換到頻率域上，以影像來說，可將其視為一多維度（二維平面圖片甚至包含 RGB 不同色域的三維圖片）的訊號，IDFT (Inverse Discrete Fourier Transform) 代表逆轉換，將樣本從頻率域轉回空間域，首先介紹轉換與逆轉換的公式：

$$\begin{aligned} F(u, v) &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \exp\left(-\frac{j2\pi um}{M}\right) \exp\left(-\frac{j2\pi vn}{N}\right) \\ &= \sum_{m=0}^{M-1} \left( \sum_{n=0}^{N-1} f(m, n) \exp\left(-\frac{j2\pi vn}{N}\right) \right) \exp\left(-\frac{j2\pi um}{M}\right) \\ f(u, v) &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F(m, n) \exp\left(\frac{j2\pi um}{M}\right) \exp\left(\frac{j2\pi vn}{N}\right) \\ &= \sum_{m=0}^{M-1} \left( \sum_{n=0}^{N-1} F(m, n) \exp\left(\frac{j2\pi vn}{N}\right) \right) \exp\left(\frac{j2\pi um}{M}\right) \end{aligned}$$

其中  $F(u, v)$  代表轉換後在  $(u, v)$  頻率上的係數， $f(m, n)$  代表圖片在空間域上  $(m, n)$  位置的像素值， $(M, N)$  分別代表圖片的高與寬有多少像素。此外不論正轉換或逆轉換，都可以分成先對圖片的每一行進行，再對每一列再轉換一次。

### 2. Fast Fourier Transform (FFT)

以 DFT 與 IDFT 的轉換公式來說，程式在計算上相當耗時，於是本篇在程式實作上參考快速傅立葉變換演算法 (Cooley-Tukey FFT algorithm) 以下簡稱 FFT。在考慮一維度變換的狀況下，FFT 最主要的想法在於原本考慮  $N$  個樣本點的離散轉換可以分解成奇數序列樣本點與偶數序列樣本點各自進行傅立葉變換後再進行線性組合，透過拆解序列的方式減少序列長度、搭配傅立葉轉換的對稱性 (Symmetry) 的方式減少計算時間，並且只要序列長度符合 2 的冪次即可持續分割，一維度 FFT 公式如下：

$$\begin{aligned} F(u) &= \sum_{n=0}^{N-1} f(n) \exp\left(-\frac{j2\pi un}{N}\right) \\ &= \sum_{m=0}^{\frac{N}{2}-1} f(2m) \exp\left(-\frac{j2\pi um}{N/2}\right) + \exp\left(-\frac{j2\pi u}{N}\right) \sum_{m=0}^{\frac{N}{2}-1} f(2m+1) \exp\left(-\frac{j2\pi u(2m)}{N/2}\right) \end{aligned}$$

### 3. Low-pass filter

考慮在頻率域上，低通濾波器 (Low-pass filter) 容許低頻訊號通過，但減弱或減少頻率高於截止頻率的訊號的通過，在影像的頻率上，高頻主要會是由噪音、輪廓及細節組成，利用低通濾波器可以達到降噪的效果。

本篇主要討論理想低通濾波器 (Ideal low-pass filter) 及高斯低通濾波器 (Gaussian low-pass filter)。理想低通濾波器主要是保留指定頻率以下的訊號，將高於指定頻率的訊號全部消除，公式如下：

$$\delta(u, v) = \begin{cases} 1, & \text{if } \sqrt{u^2 + v^2} \leq D_0 \\ 0, & \text{other} \end{cases}$$

其中  $D_0$  代表指定的頻率。在對影像從空間域轉換至頻率域並且進行集中轉移 (centralized shift) 後，以中心代表原點，僅保留距離小於  $D_0$  的低頻。而在高斯低通濾波器設計上並非像理想濾波器以一閾值 (Threshold) 分割保不保留，反而採用高斯分布的指數遞減函數來代表隨著距離越來越遠，保留的訊息也越來越小，公式如下：

$$\delta(u, v) = \exp\left(-\frac{u^2 + v^2}{2\sigma^2}\right)$$

其中  $\sigma$  代表遞減速度，當  $\sigma$  越大便會使得  $\delta(\dots)$

### 4. High-pass filter

高通濾波器 (High-pass filter) 與低通濾波器恰好相反，高通濾波器容許高頻訊號通過，但減弱或減少頻率低於截止頻率的訊號的通過，主要可以利用高通濾波器保留輪廓、細節的訊息。

本篇主要討論理想高通濾波器 (Ideal high-pass filter) 及巴特沃斯高通濾波器 (Butterworth high-pass filter)。理想高通濾波器公式如下：

$$\delta(u, v) = \begin{cases} 1, & \text{if } \sqrt{u^2 + v^2} \geq D_0 \\ 0, & \text{other} \end{cases}$$

其中  $D_0$  代表指定的頻率。

而在巴特沃斯高通濾波器設計上以遞減函數來代表隨著距離越來越遠，保留的訊息也越來越小，公式如下：

$$\delta(u, v) = 1 - \frac{1}{1 + (\sqrt{u^2 + v^2}/D_0)^{2n}}$$

其中  $D_0$  與  $n$  皆為指定參數， $D_0$  類似於理想濾波器的閾值，而當  $n$  越來越大時，巴特沃斯高通濾波器會越近似理想高通濾波器。

## 5. Denoising filter

原本在空間域上，降噪的濾波器通常以卷積的計算方式來達成，但在有了頻率分析後其實降噪也可以轉換到頻率上進行。影像上的噪音其實可以考慮成影像受到了衰變，衰變在空間域上可以考慮影像是受到一特殊的 kernel 卷積，而在頻率域上便是原影像的頻率乘上了特殊 kernel 的頻率，衰變的過程可以以公式表示如下：

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

其中  $g, h, f, \eta$  代表在空間域上的衰變結果、特殊的 kernel、原始影像及噪音，而  $G, H, F, N$  則代表  $g, h, f, \eta$  轉換到頻率域上的結果， $*$  代表卷積。

本篇主要討論逆濾波器 (Inverse filter)、維納濾波器 (Wiener filter)、BM3D 及導向濾波器 (Guided filter)。

### 5.1. 逆濾波器 (Inverse filter)

逆濾波器的想法很簡單，在特殊 kernel 及其頻率  $H(u, v)$  已知的狀況下可以考慮直接相除， $\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)}$ ，但在實作上會出現許多問題，第一：在通常情況下難以得知特殊 kernel 及其頻率  $H(u, v)$  為何，第二： $N(u, v)$  並未被處理。根據上述的兩個原因，逆濾波器很難轉換出不錯的結果。

### 5.2. 維納濾波器 (Wiener filter)

維納濾波器主要使用於因加性噪音和模糊而退化的圖像。在計算上需要假設信號和噪聲過程是二階平穩，估計式如下：

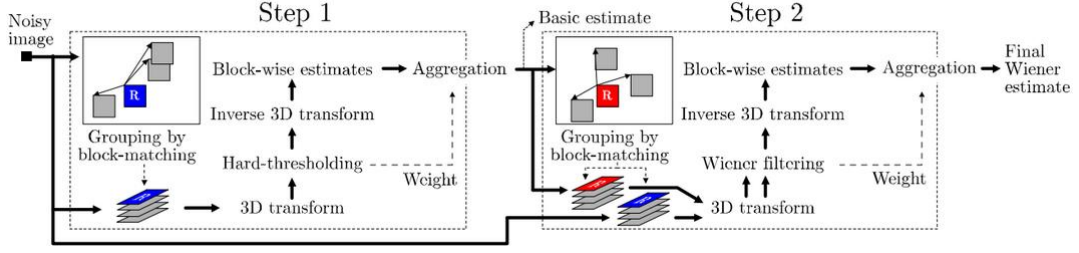
$$\hat{F}(u, v) = W(u, v)G(u, v)$$

$$W(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + K(u, v)}$$

其中  $H^*(u, v)$  代表  $H(u, v)$  的共軛 (Conjugate)， $K(u, v)$  為一調整參數，當  $K=0$  代表  $W(u, v)$  退化為逆濾波器。在維納濾波器中有個好處是即使  $H(u, v)$  指定的並非造成衰變的真實濾波器，輸出的結果也不會太糟糕，通常可以透過調整  $K$  的大小來改善。

### 5.3.BM3D

主要想法可以認為是空間域的降噪方法非局部平均濾波器 (Non-local means filter) 與頻率域進行結合，其主要可分為三個步驟，即首先對每個參考塊進行相似塊匹配 (Block-matching) 並分別得到一個三維的組合，然後對其進行協同變換和濾波 (3D-Transform)，最後對各個參考塊對應組合的濾波結果進行整合 (Aggregation)，從而得到最終的降噪結果。在此之上，為了進一步改善圖像的品質，BM3D 實際進行了兩次降噪，即將以上三個步驟再重複了一遍，但是具體的塊匹配標準、濾波方式以及整合權重等等會有一些區別，主要流程如下圖：



### 5.4.導向濾波器 (Guided filter)

導向濾波的名字由來，在於其演算法中，對於一張輸入圖  $p$ ，我們還額外需要提供另一張引導圖  $I$ ，才有辦法做濾波。導向濾波首先假設輸出圖  $q$  與引導圖  $I$  之間有局部線性的關係，公式可表示如下：

$$q_i = aI_i + b$$

其中  $a, b$  為需求解的參數。而輸出圖  $q$  同時也等同輸入圖  $p$  去除雜訊  $n$  後的結果，公式如下：

$$q_i = p_i - n_i$$

因此我們可以得到一組聯立方程式，最後便可對  $n$  以最小平方法並施加正則化 (Regularization) 的方式來求解，求解公式如下：

$$E(a_k, b_k) = \sum_{i \in w_k} [(a_k I_i + b_k - p_i)^2 + \epsilon a_k^2]$$

其中  $w_k$  代表在指定半徑的指標 (index)  $\epsilon$  正則化的指定參數。

演算法流程如下：

**Algorithm 1. Guided Filter.**

**Input:** filtering input image  $p$ , guidance image  $I$ , radius  $r$ , regularization  $\epsilon$

**Output:** filtering output  $q$ .

- 1:  $\text{mean}_I = f_{\text{mean}}(I)$   
 $\text{mean}_p = f_{\text{mean}}(p)$   
 $\text{corr}_I = f_{\text{mean}}(I * I)$   
 $\text{corr}_{Ip} = f_{\text{mean}}(I * p)$
- 2:  $\text{var}_I = \text{corr}_I - \text{mean}_I * \text{mean}_I$   
 $\text{cov}_{Ip} = \text{corr}_{Ip} - \text{mean}_I * \text{mean}_p$
- 3:  $a = \text{cov}_{Ip} / (\text{var}_I + \epsilon)$   
 $b = \text{mean}_p - a * \text{mean}_I$
- 4:  $\text{mean}_a = f_{\text{mean}}(a)$   
 $\text{mean}_b = f_{\text{mean}}(b)$
- 5:  $q = \text{mean}_a * I + \text{mean}_b$

/\*  $f_{\text{mean}}$  is a mean filter with a wide variety of  $O(N)$  time methods. \*/

## 6. Discrete Cosine Transform (DCT)

離散餘弦變換 (Discrete Cosine Transform; DCT) 是與傅立葉變換相關的一種變換，類似於離散傅立葉變換，但是只使用實數。DCT 有 8 種標準類型，其中 4 種最為常見，以下陳列常見的第二型及其逆離散餘弦變換 (Inverse Discrete Cosine Transform; IDCT)，變換公式如下：

$$\begin{aligned}
 F(u, v) &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \alpha_M(u) \alpha_N(v) \cos\left(\frac{\pi(2m+1)u}{2M}\right) \cos\left(\frac{\pi(2n+1)v}{2N}\right) \\
 &= \sum_{m=0}^{M-1} \left( \sum_{n=0}^{N-1} f(m, n) \alpha_N(v) \cos\left(\frac{\pi(2n+1)v}{2N}\right) \right) \alpha_M(u) \cos\left(\frac{\pi(2m+1)u}{2M}\right) \\
 f(u, v) &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F(m, n) \alpha_M(u) \alpha_N(v) \cos\left(\frac{\pi(2u+1)m}{2M}\right) \cos\left(\frac{\pi(2v+1)n}{2N}\right) \\
 &= \sum_{m=0}^{M-1} \left( \sum_{n=0}^{N-1} F(m, n) \alpha_N(v) \cos\left(\frac{\pi(2n+1)v}{2N}\right) \right) \alpha_M(u) \cos\left(\frac{\pi(2m+1)u}{2M}\right)
 \end{aligned}$$

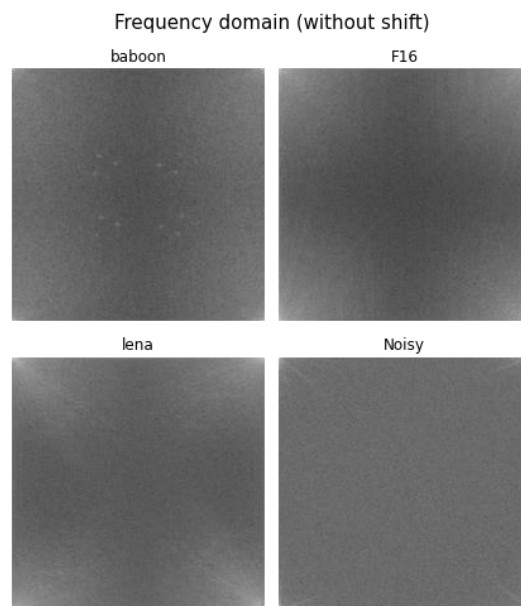
其中  $\alpha_M(0) = M^{-\frac{1}{2}}, \alpha_N(0) = N^{-\frac{1}{2}}, \alpha_M(k) = \sqrt{2/M}, \alpha_N(k) = \sqrt{2/N}$  for  $k > 0$

## Result

本節主要以上述的方法搭配使用一開始提及的 4 張不同測試圖片來呈現結果。

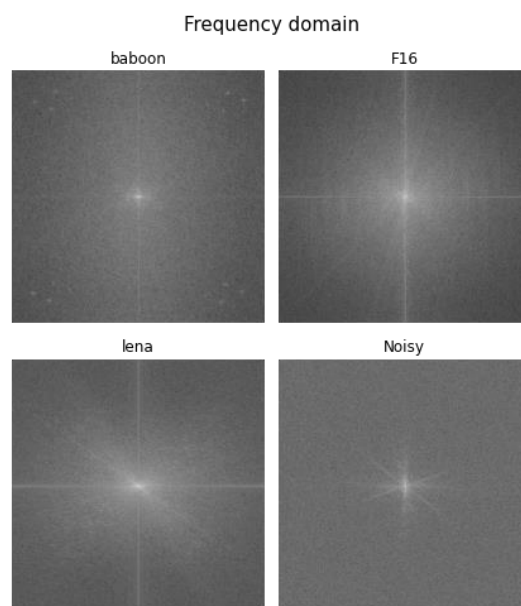
### 1.1. Draw the frequency response without shifting.

下方為視覺化 4 張測試圖片利用 FFT 所轉換的結果。在經由 FFT 轉換後取絕對值代表該頻率的強度，再將其取自然對數 (Natural logarithm)，最後再單位化到  $[0,1]$  後乘以 255。在沒有進行集中轉移 (centralized shift) 下，多餘 4 個角落有較大值。



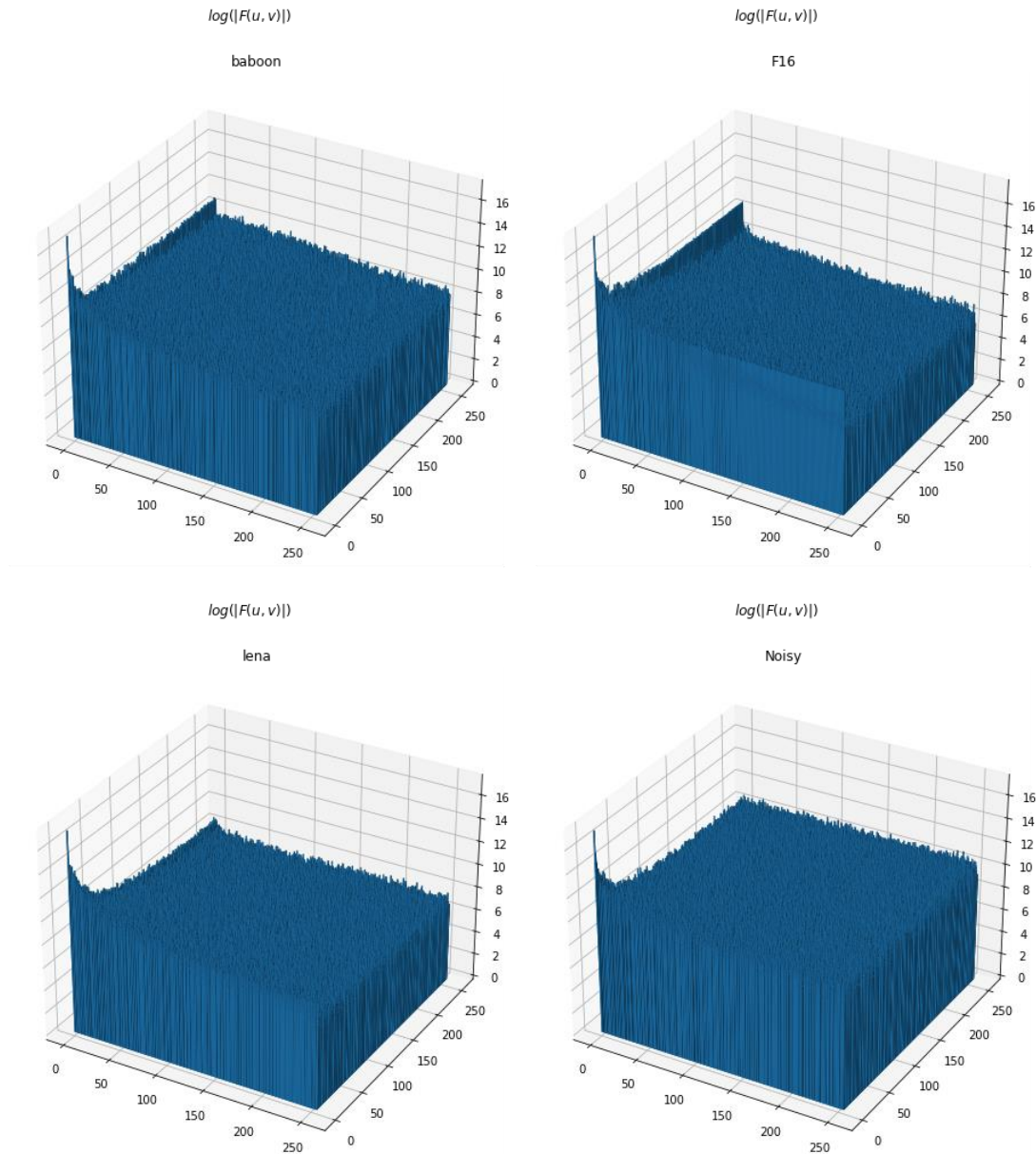
### 1.2. Show the centralized result based on (1.1).

本題結果為上一小題的轉換結果在額外考慮集中轉移 (centralized shift) 的樣貌。集中轉移以圖片中心切割 4 個象限後，二、四象限交換以及一、三象限交換，透過此作法變可將低頻的資訊繪製於圖片中央。



### 1.3. Plot the histogram of the spectrum.

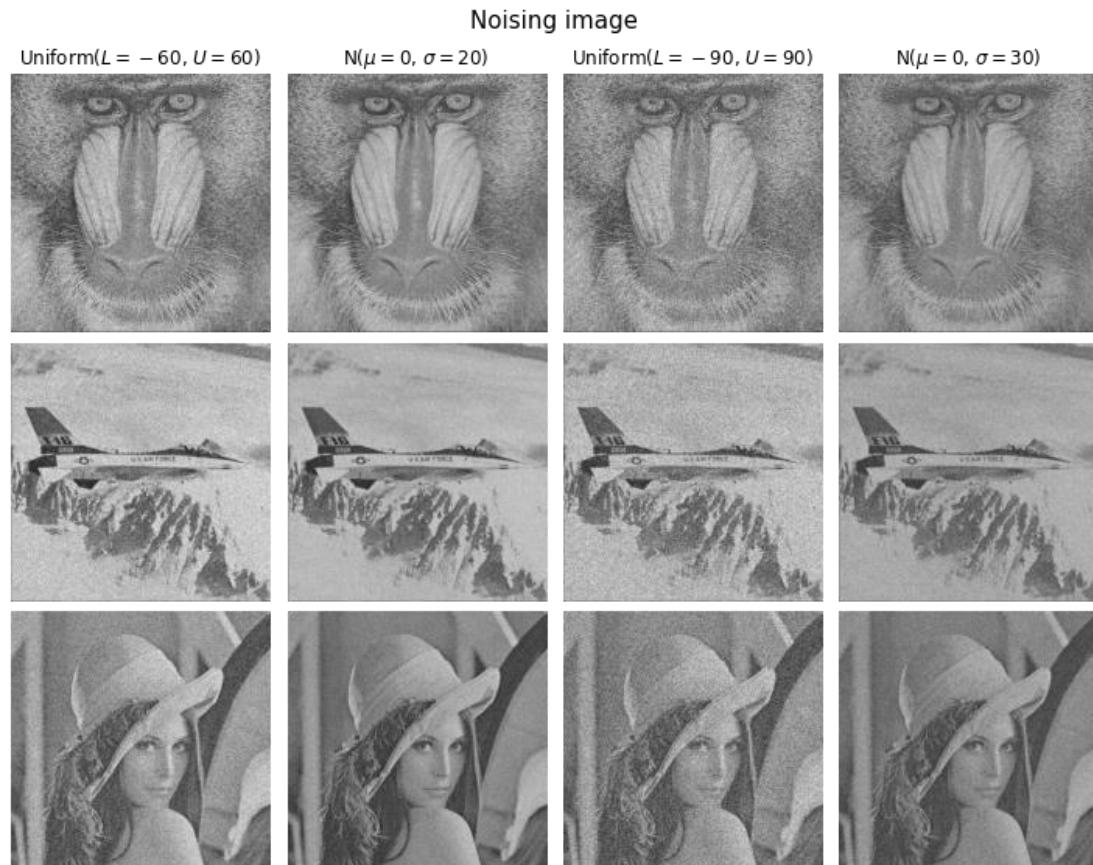
下圖為不同測試圖片在經由 FFT 後將其值取絕對值在取自然對數的柱狀圖，原本的圖片皆為  $512 \times 512$ ，因此在頻率上只取  $[:256,:256]$  來作呈現，藉此我們可以發現在 noisy.raw 的高頻部分的強度確實比其他測試圖片高，而 fl6.raw 可以在下圖發現的低頻與高頻下降速度很快。





2.1. Make a Random Noise generator (uniform and Gaussian with specific (mean/var) to add the noise to the original images in Spatial Domain.

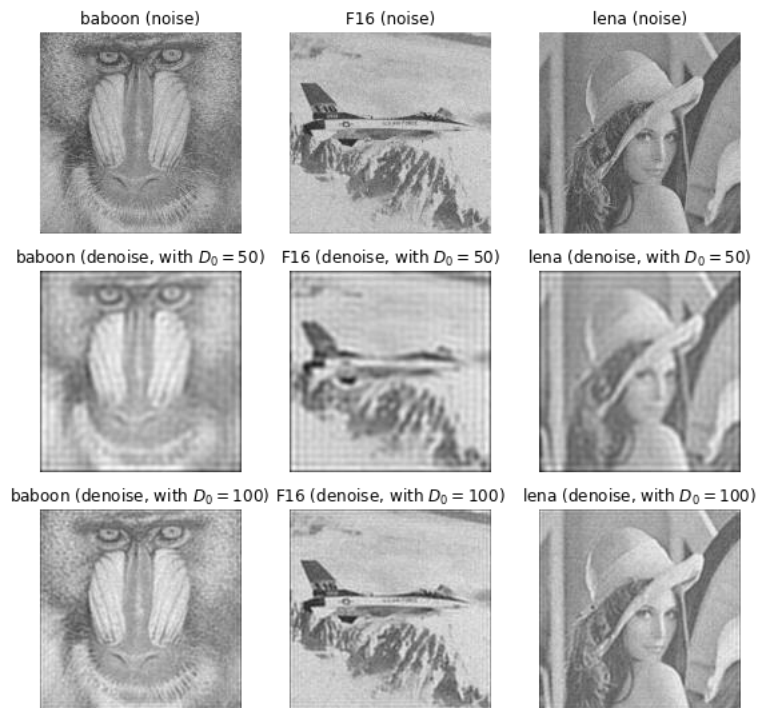
下圖呈現為測試圖片加入不同的噪音的結果，由左至右分別代表加入了不同類型的噪音，主要考慮兩種不同的機率分配，分別為均勻分配以及高斯分配在不同參數下對原圖的影像。在後面兩個小題當中考慮均勻分配 ( $Unif(L = -60, U = 60)$ ) 及高斯分配 ( $Gaussian(\mu = 0, \sigma = 20)$ ) 進行測試。



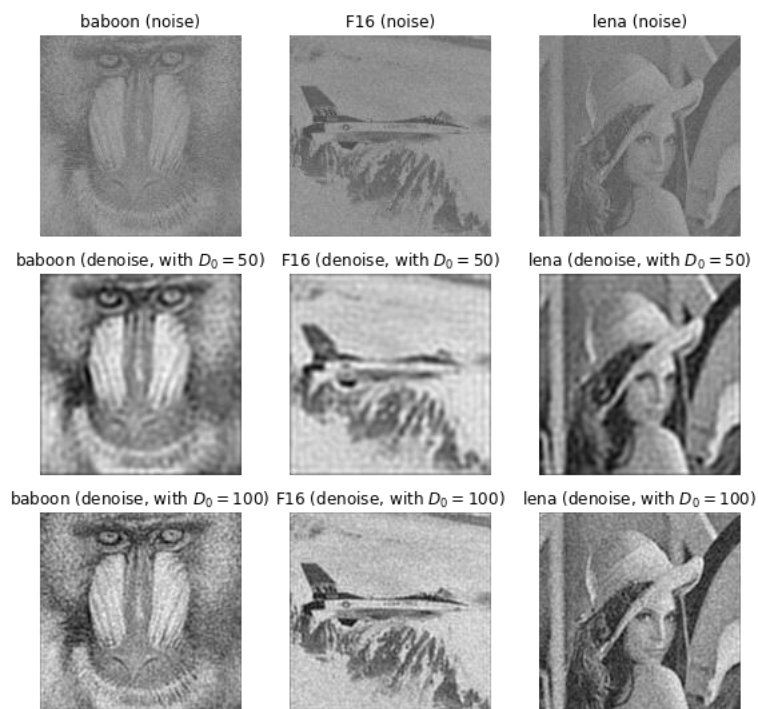
2.2. Apply the ideal low-pass filter to obtain the denoised results with different settings.

下圖為使用理想低通濾波器在不同  $D_0$  情況下對具有均勻分配、高斯分配均勻分配噪音的圖片進行處理的結果，當  $D_0$  較大時能保有較多細節特徵。

Ideal low-pass filter with Uniform( $L = -60, U = 60$ ) noise

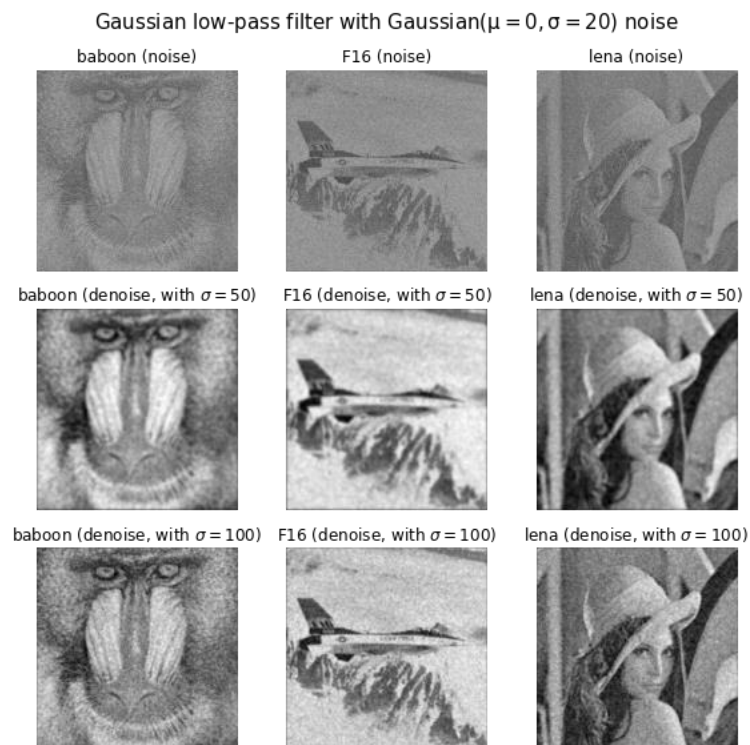
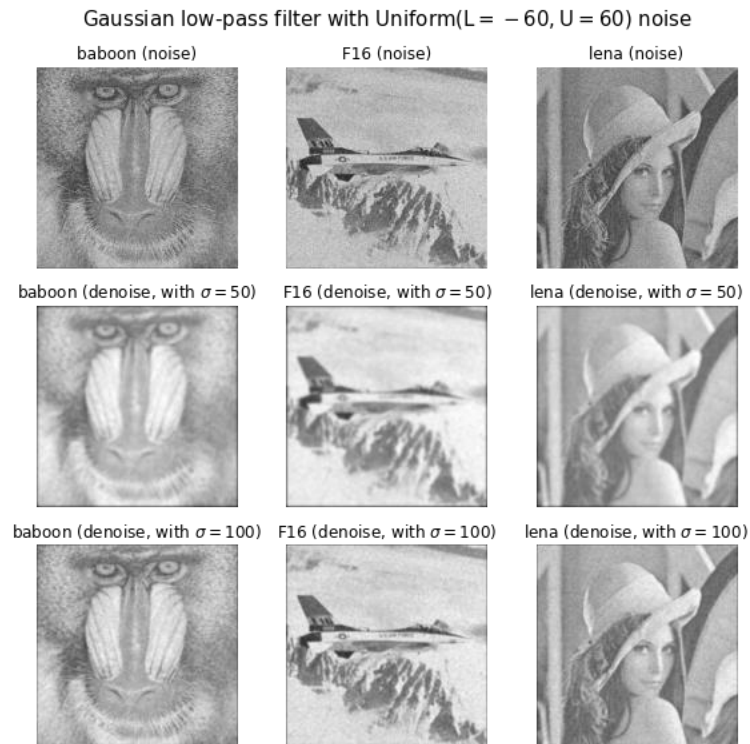


Ideal low-pass filter with Gaussian( $\mu = 0, \sigma = 20$ ) noise



2.3. Apply the Gaussian low-pass filter to obtain the denoised results with different settings.

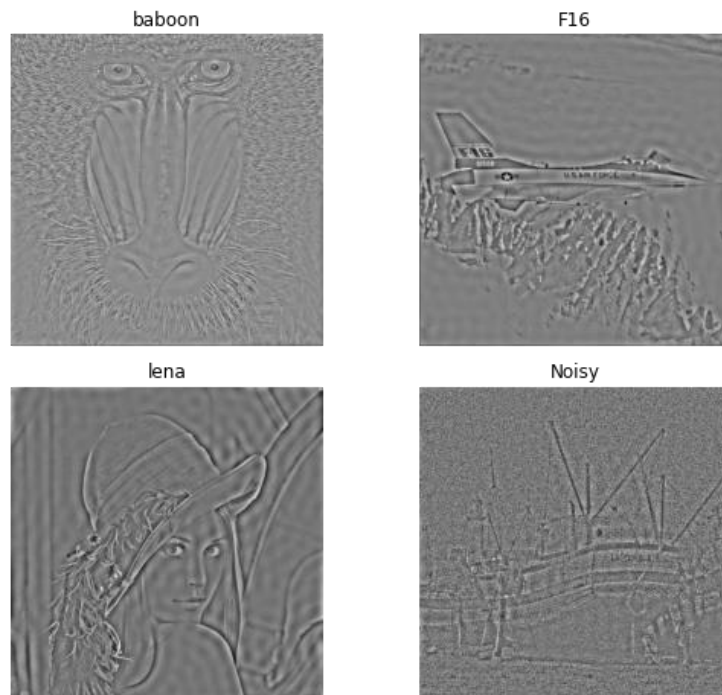
下圖為使用高斯低通濾波器在不同  $\sigma$  情況下對具有均勻分配、高斯分配均勻分配噪音的圖片進行處理的結果，當  $\sigma$  較大時能保有較多細節特徵。



3.1. Please use the ideal and Butterworth high-pass filters. Make a comparison between the frequency high-pass filter and the Laplacian filter.

下圖為在頻率域上使用理想高通濾波器及巴特沃斯高通濾波器對不同測試圖片的濾波結果以及在空間域上使用拉普拉斯濾波器設定  $\text{kernel size}=3 \times 3$  的結果。由此可發現高頻的部分為圖片當中的紋理、輪廓以及噪音。另外，比起拉普拉斯濾波器，另外兩者在頻率域上操作後的結果表現更為清晰，可見在頻率域上利用其物理意義能處理的更佳。

Ideal high-pass filter ( $D_0 = 20$ )



Butterworth high-pass filter ( $D_0 = 20, n = 2$ )

baboon



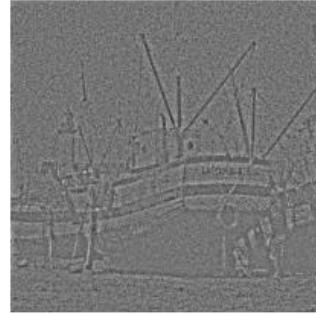
F16



lena

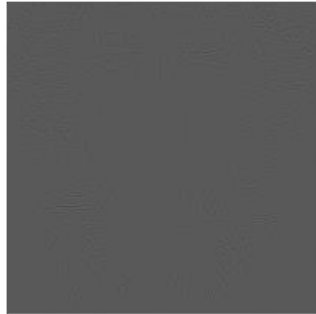


Noisy



Laplacian filter ( $3 \times 3$ )

baboon



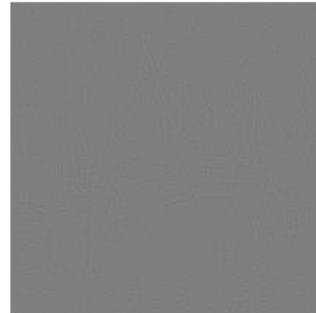
F16



lena



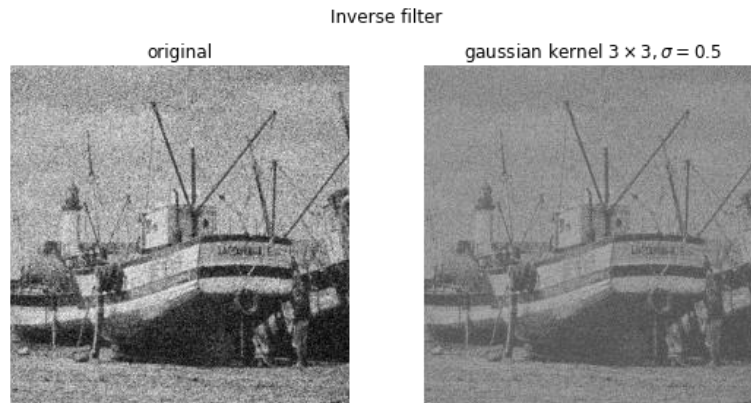
Noisy



## 4. Image denoising

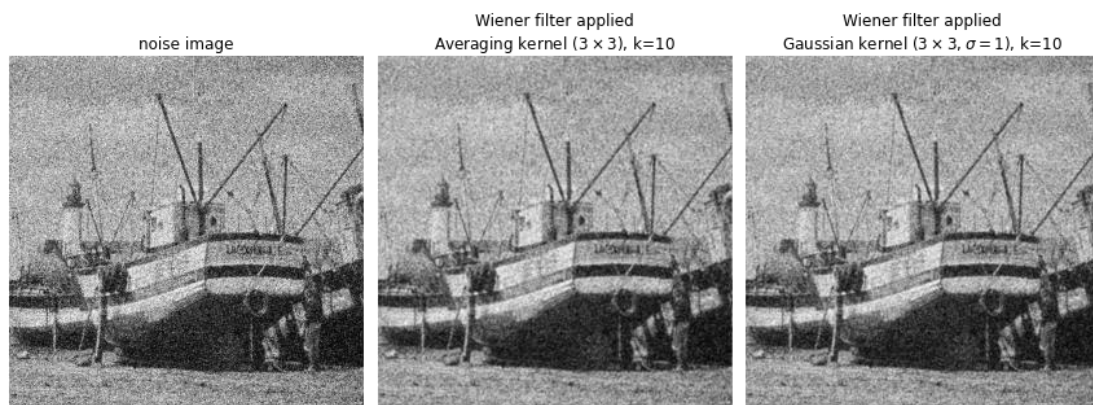
### 4.1. Inverse filter

在逆濾波器上嘗試了許多不同種類的平滑核及其設定，效果都相當不好，整張圖片幾乎會全毀變成灰色，猜想或許在原圖上可能沒有  $H(u, v)$  造成的衰變，只有  $N(u, v)$  的噪音，因此逆濾波器無法做出好的結果。



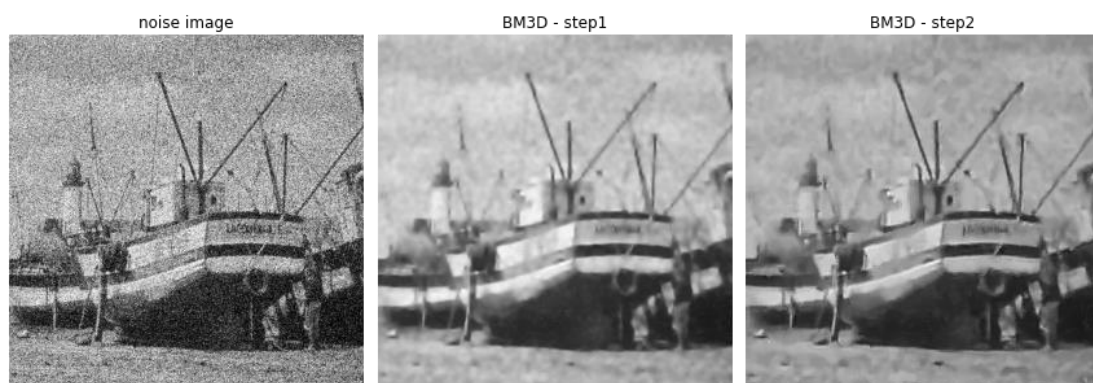
### 4.2. Wiener filter

下圖為使用維納濾波器在考慮不同平滑核 (smooth kernel) 的結果，結果都不會像是逆濾波器那樣糟糕，至少都還有些降噪的效果存在。



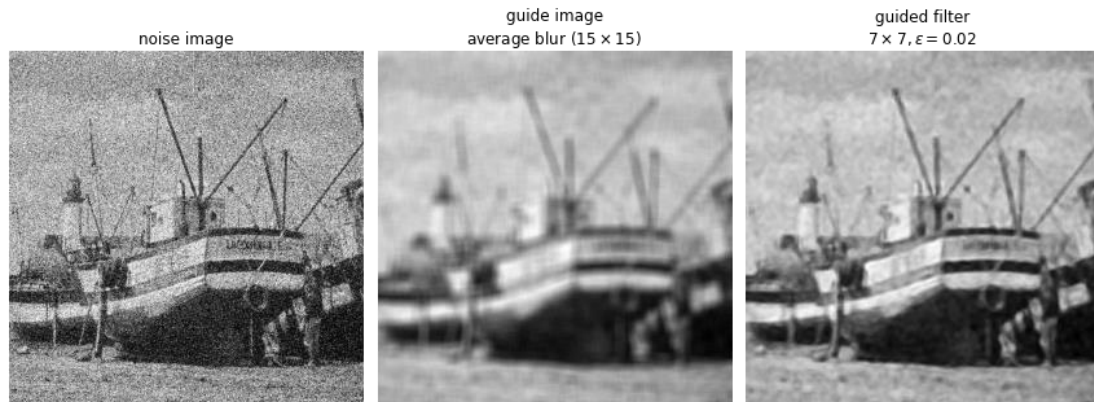
### 4.3. BM3D

在 BM3D 上有不錯的降噪結果，但實作中計算 step1 與 step2 共耗時 78 秒與 177 秒，計算成本較其他方法高出許多。



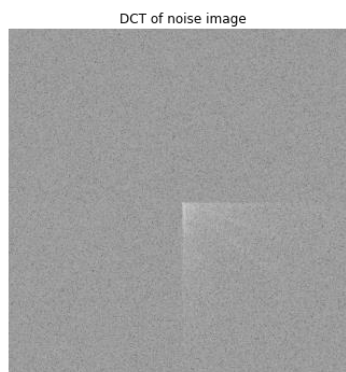
#### 4.4. Guided filter

在導向濾波器當中，根據我們要降噪的目的，給予的導圖為其經由  $15 \times 15$  kernel size 利用平均濾波器糊化，最右方為使用導向濾波器使用  $7 \times 7$  kernel size 正則化的參數  $\epsilon = 0.02$  的結果。



#### 5. DCT as the image restoration domain.

下圖為 noisy.raw 的圖片在經過 DCT 轉換以及 centralized shift 後的結果，藉此便可套用前面討論的一些低通濾波器來進行降噪。



下圖為對 DCT 轉換的結果利用高斯低通濾波器在  $\sigma = 30$  及  $\sigma = 60$  的濾波結果。

