

1. Fourier Transform

傅立葉轉換將原本的圖片轉為用頻率表示的圖片，一維訊號轉換公式如下

$$X_k = \sum_{n=0}^{N-1} x_n \times e^{-i2\pi kn/N}$$

式子中 X_k 為頻率空間中基底訊號 $e^{-i2\pi kn/N}$ 的係數 x_n 為原始訊號，逆運算為

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \times e^{i2\pi kn/N}$$

如果直接使用這個式子對資料做轉換會使用太多時間，所以改成使用快速傅立葉轉換，式子改為

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} \times e^{-i2\pi km/(N/2)} + e^{-i2\pi k/N} \sum_{m=0}^{N/2-1} x_{2m+1} \times e^{-i2\pi km/(N/2)}$$

使用此式子可以將原本的訊號拆為奇數項和偶數項，基底式子 $e^{-i2\pi km/(N/2)}$ 在兩項中皆相同，運算時可以減少計算基底式子的時間，且偶數項

$$\sum_{m=0}^{N/2-1} x_{2m} \times e^{-i2\pi km/(N/2)}$$

和基數項中連加的部分

$$\sum_{m=0}^{N/2-1} x_{2m+1} \times e^{-i2\pi km/(N/2)}$$

可以視為是另一個訊號的傅立葉轉換，將它在拆解為奇數項和偶數項，如此遞迴拆解大量減少要計算的基底式子部分。

上述方法是 k 固定時加快的方法，也就是在計算某個頻率基底 $e^{i2\pi kn/N}$ 的係數時加快的方法，另一個加速方法是對 k 做加速，以下為加速的式子

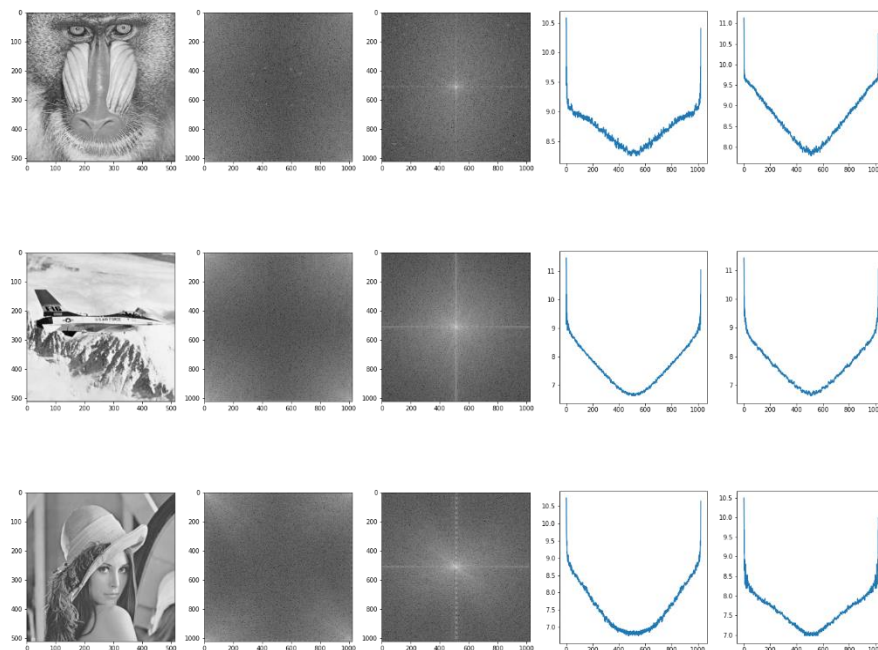
$$\begin{aligned} & X_{k+N/2} \\ &= \sum_{m=0}^{N/2-1} x_{2m} \times e^{-i2\pi(k+N/2)m/(N/2)} \\ & \quad + e^{-i2\pi(k+N/2)/N} \sum_{m=0}^{N/2-1} x_{2m+1} \times e^{-i2\pi(k+N/2)m/(N/2)} \end{aligned}$$

$$= \sum_{m=0}^{N/2-1} x_{2m} \times e^{-i2\pi km/(N/2)} - e^{-i2\pi k/N} \sum_{m=0}^{N/2-1} x_{2m+1} \times e^{-i2\pi km/(N/2)}$$

這個式子與計算 X_k 的式子中間只差一個負號，利用這個式子可以先計算訊號前半段轉換後的值，再利用計算前半段式子中的乘上一個負號得到訊號後半段的值，這樣可以大樣減少計算基底 $e^{i2\pi kn/N}$ 的時間，但這樣做需要資料為 2 的冪次方倍，所以會將訊號做 padding 得到一個增大的訊號且訊號量為 2 的冪次方倍。

最後將式子推到二維資料，實作時只要將圖將資料的每個列先做傅立葉轉換，在對每個 **column** 做一次即可得到二維的傅立葉轉換資料。

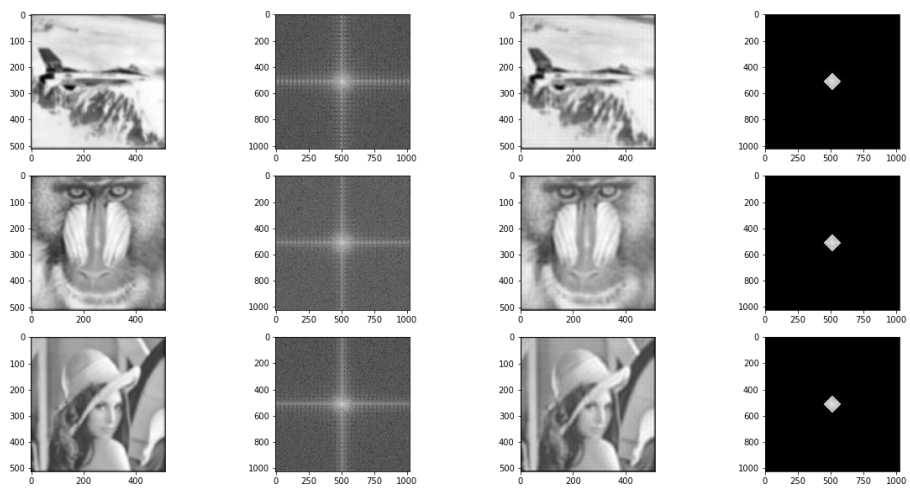
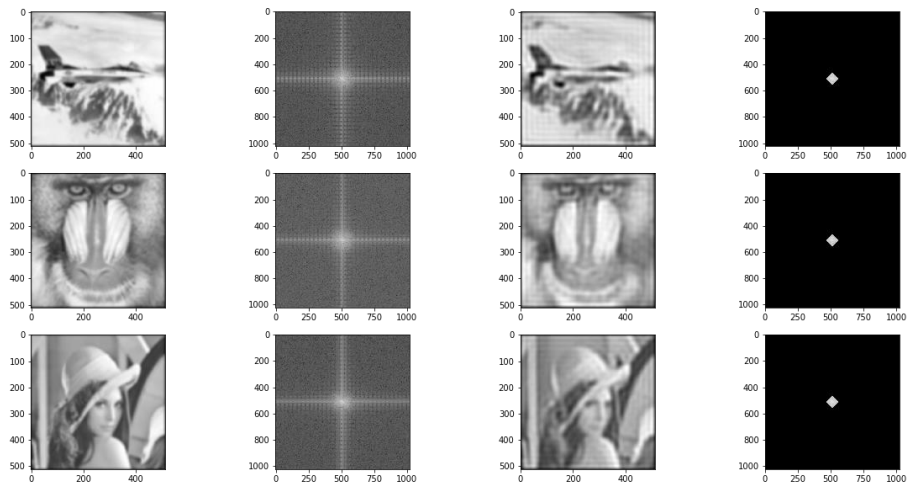
以下為實作結果

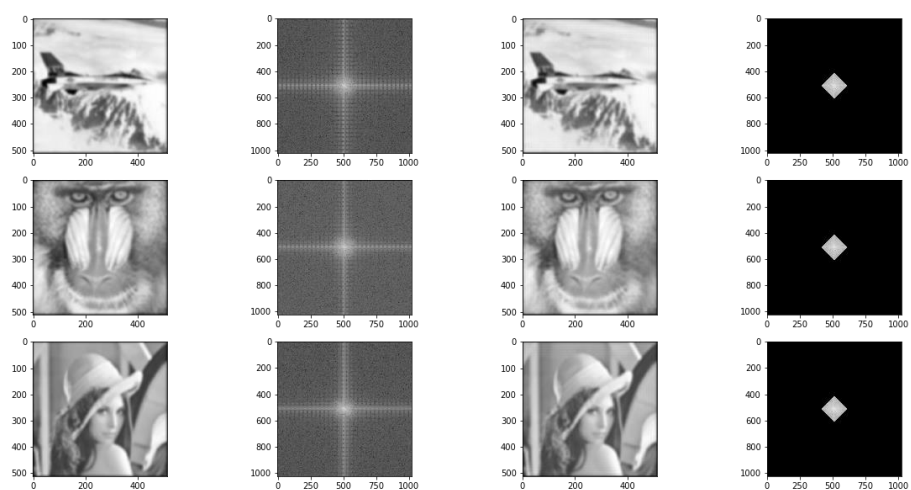


最左邊為原圖，第二行為傅立葉轉換後的訊號，第三行為將第二行的圖片訊號切四等分後將四個角的訊號轉到最中間，史最低頻的部分集中在中間，第四行的橫軸為圖片 **row** 的頻率基底，縱軸為該基底的係數，第五行圖片為原圖片 **column** 頻率基底的係數，第四行和第五行的圖中間為高頻部分，兩邊為低頻，從圖片可看出猴子的圖在高頻部分有最大的係數，高頻部分最低的是飛機的圖，這代表猴子的圖有較複雜的顏色變化，飛機的圖顏色變化則較少。

2. Low-pass filter

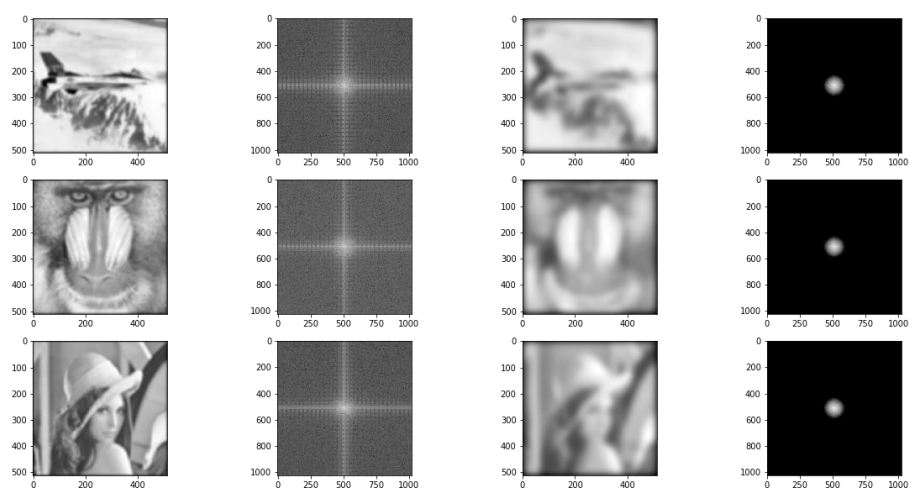
底下為使用 ideal filter 製作的 low pass filter 結果

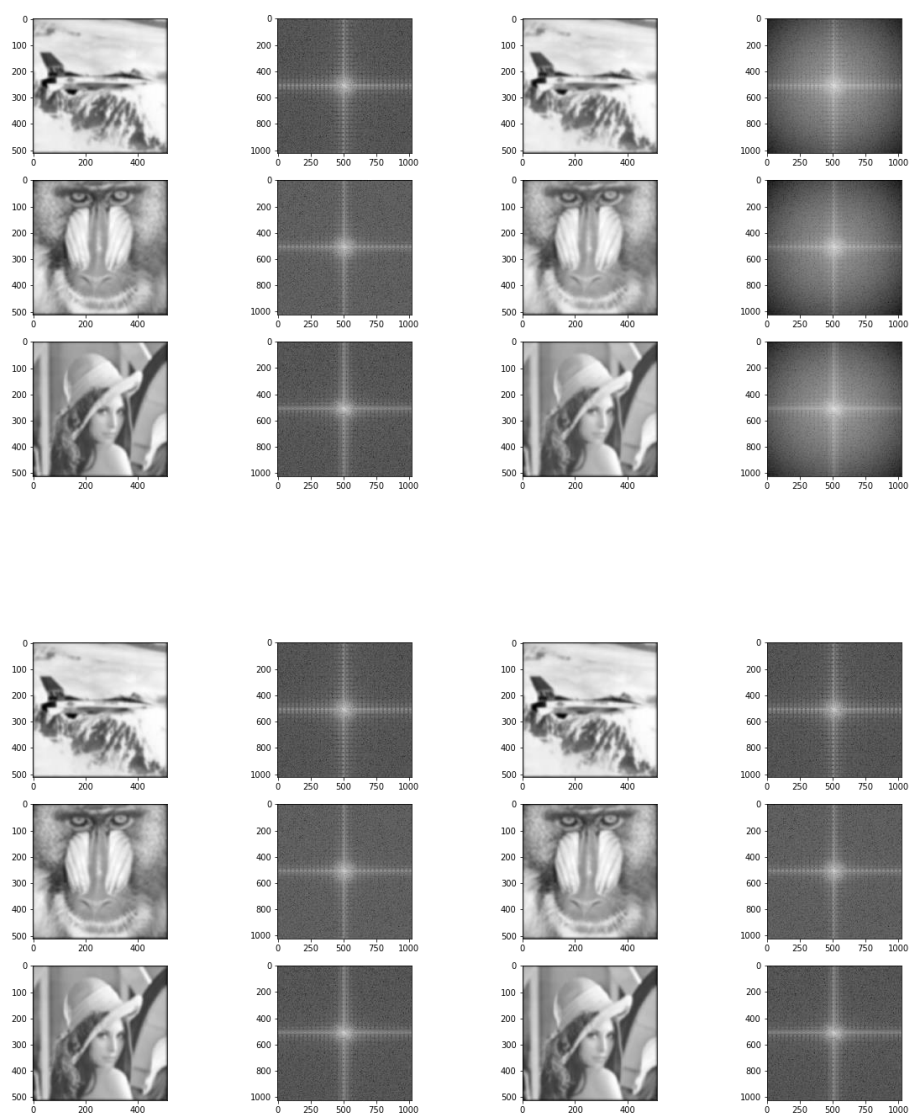




最左邊為經過高斯模糊後的圖，第二行為原圖的傅立葉轉換，第四行為原圖傅立葉轉換去除高頻的部分，第三行為轉回來的圖。

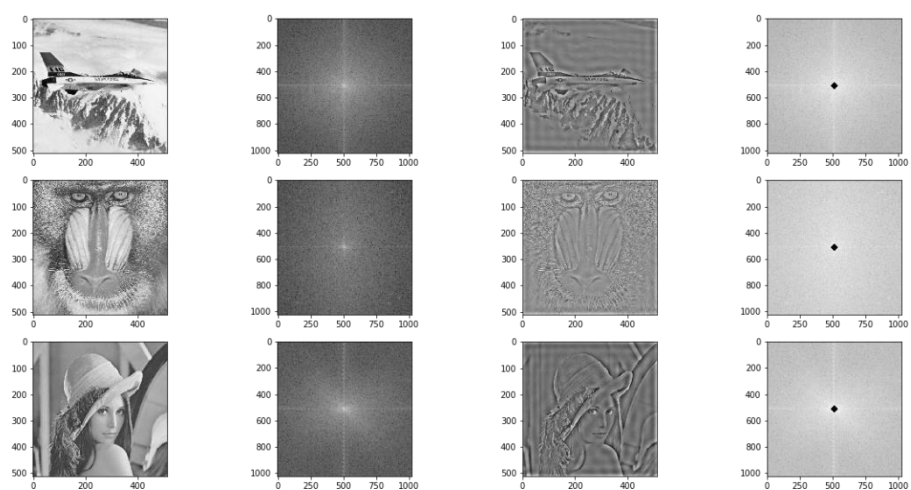
以下為使用 gaussian filter 製作的 low pass filter



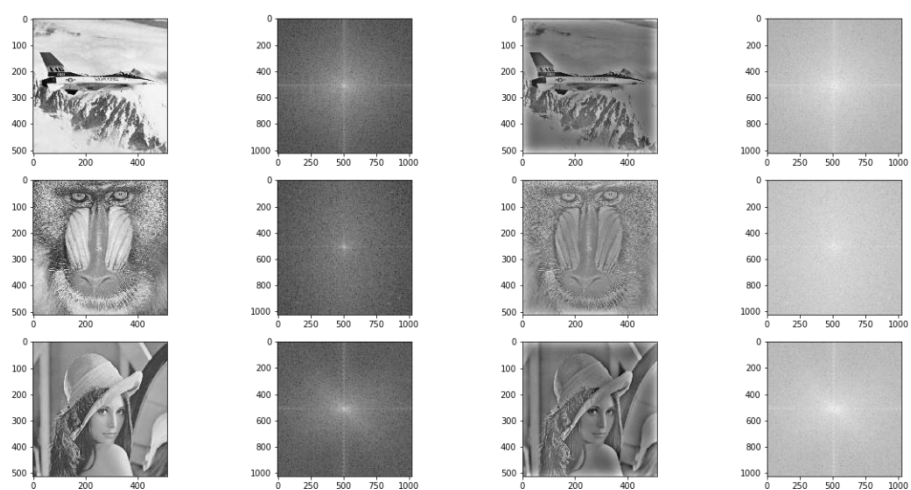


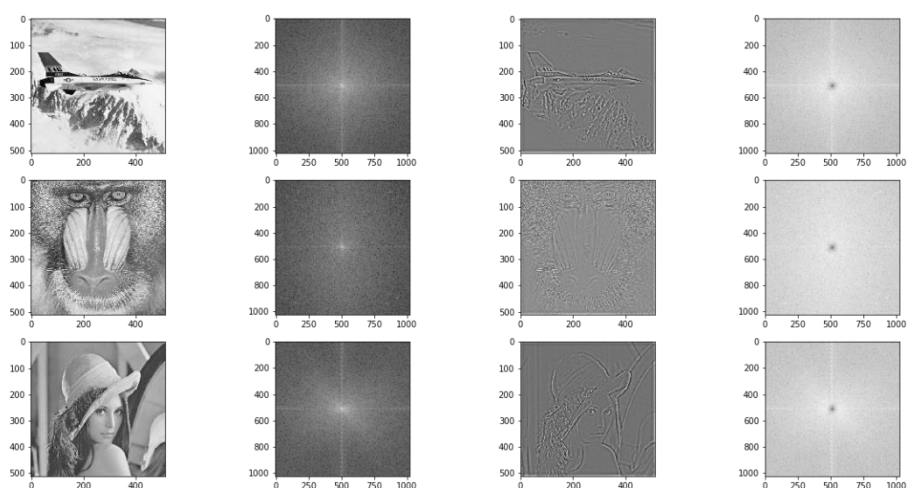
3. high-pass filter

以下為使用 ideal filter 的 high pass filter



以下為使用 butterworth filter 製作的 high pass filter

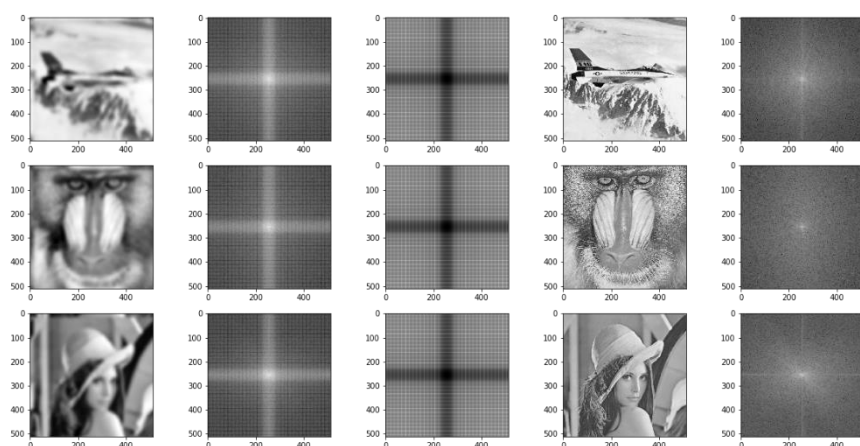




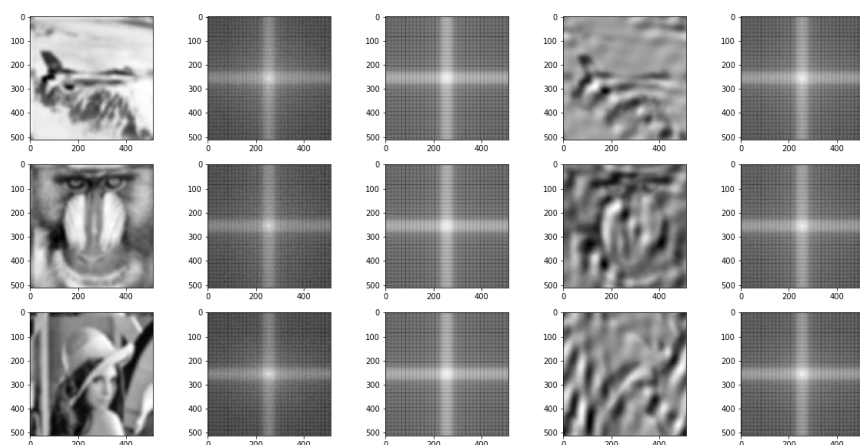
與使用 Laplacian filter 的結果比起來，這個表現得更加細緻，使用 convolution 製作時邊緣檢測會同時受到圖片縱軸部分和橫軸部分的影響，若縱軸的部分變化剛好將橫軸的部分變化抵銷掉則可能無法檢測到，使用頻率則可以較不受這點影響。

4. Image denoising

底下為使用 inverse filter 的結果

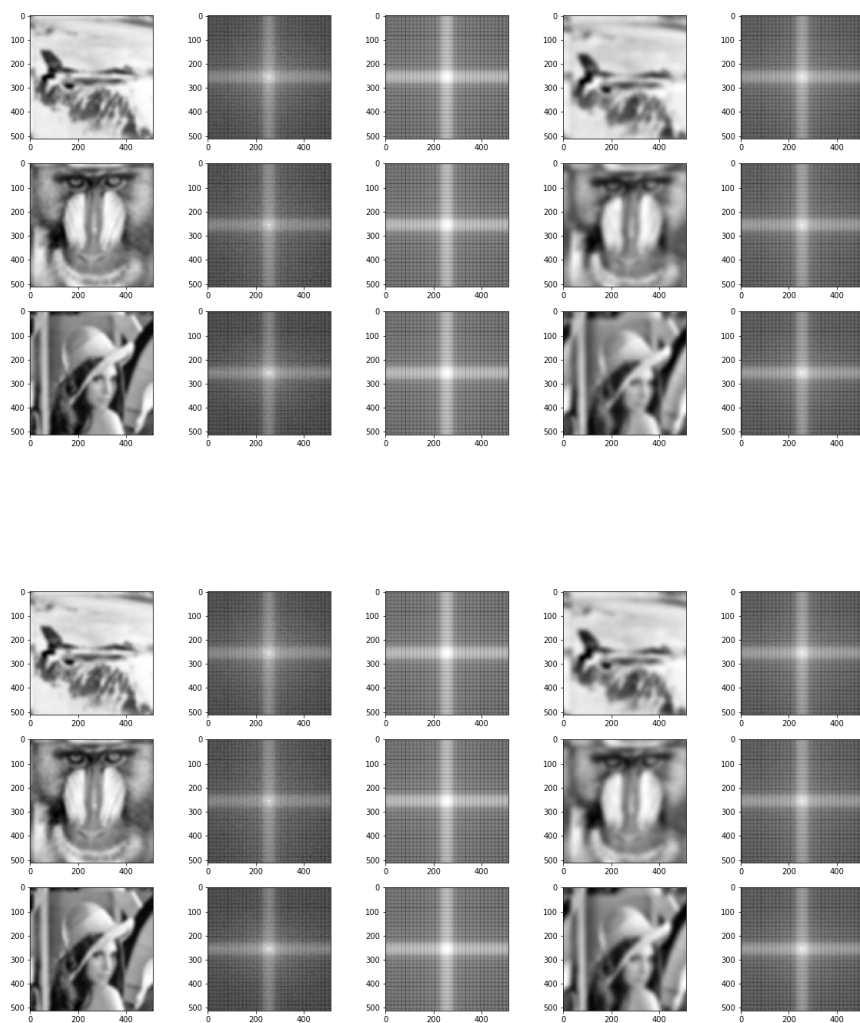


第一行為經過模糊後的圖片，第二行為它的傅立葉轉換，第三行的圖為猜測原圖片受到某個高斯 filter 的攻擊使圖片變模糊，該高斯 filter 的傅立葉空間轉換，第四張圖為修復後的圖，此實驗結果為正確猜測被攻擊的圖片是受到什麼高斯 filter 攻擊後可以推回原來的圖，但假如猜測錯誤則可能會得到下圖結果，



結果會很差，所以製作 Wiener filter 修正這項問題。

底下為使用 Wiener filter 的結果

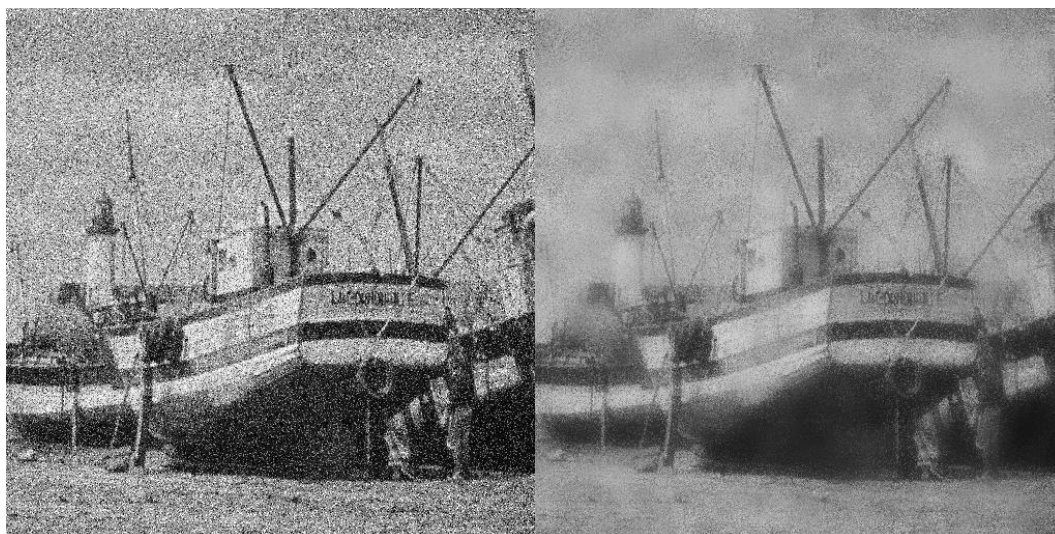


使用此方法即使沒有正確猜測受到的攻擊也能整雜訊強度對圖片的影響程度還原圖片。

以下為使用 BM3D 的結果

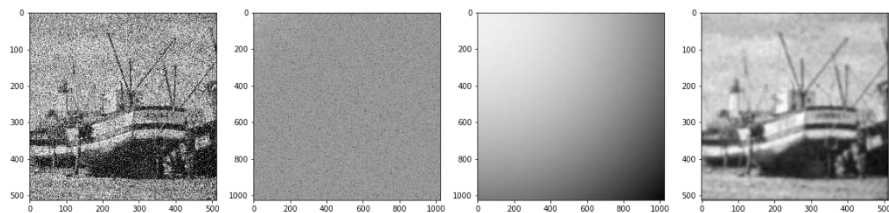


以下為 Guided filter 的結果



5. DCT

以下為 DCT 的結果



最左邊為原圖，第二行為 DCT 的頻率圖，第三行為使用 gaussian low pass filter 處理過後的頻率圖，最後一張為 denoising 後的圖。