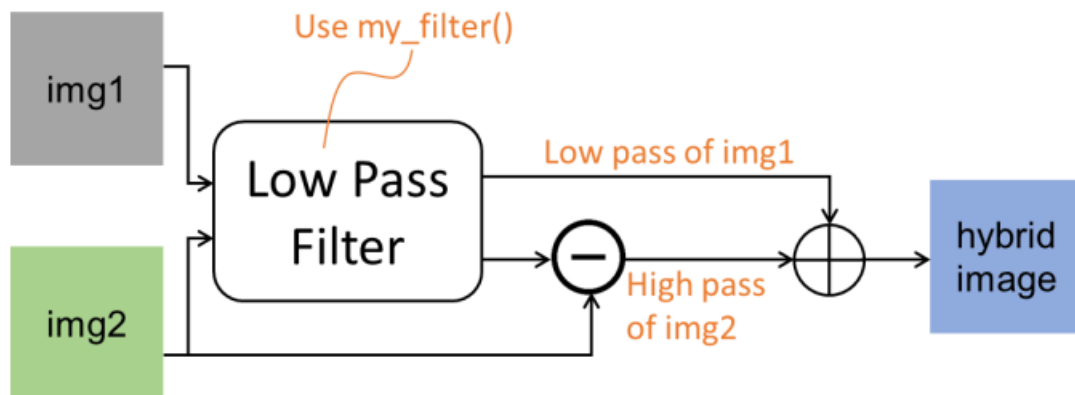


- Lab Objective

實做 Hybrid image 並探討其背後的原理以及解決一些實作中遇上的問題。

- Design Implementation

我們實作 Hybrid image 的步驟如下圖：



這裡所使用的 lowpass filter 是一個 Gaussian filter，我們需要決定他的 cutoff frequency。

我們保留其中一張圖片的 low frequency part 以及另一張圖片的 high frequency part 並將他們疊加，就能產生 Hybrid image。下面是我有嘗試的幾種組合的成果

A. Bicycle (low freq) + Motorcycle (high freq)

Original

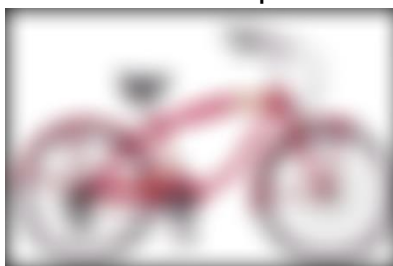


Low freq



High freq

Hybrid





B. Bird (Low Freq)+ Plane (High Freq)

Original



Low freq

High freq

Hybrid



C. Dog (Low Freq) + Cat (High Freq)

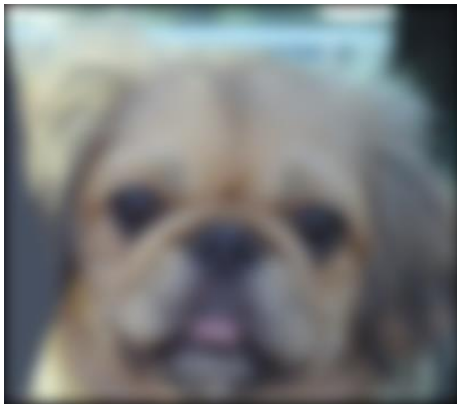
Original



Low Freq

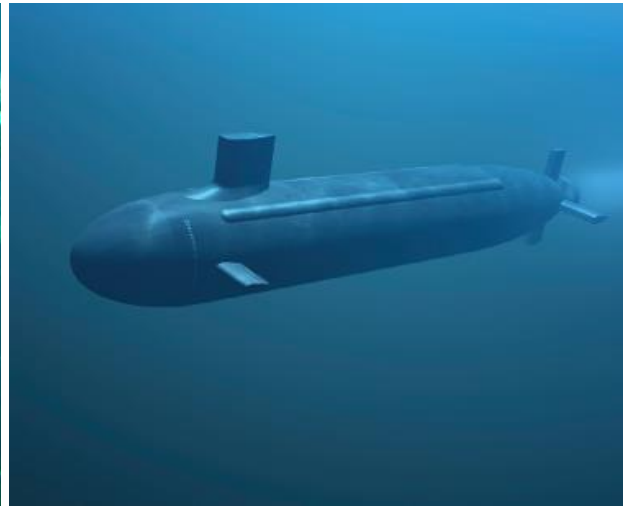
High Freq

Hybrid



D. Fish(low freq) + Submarine(high freq)

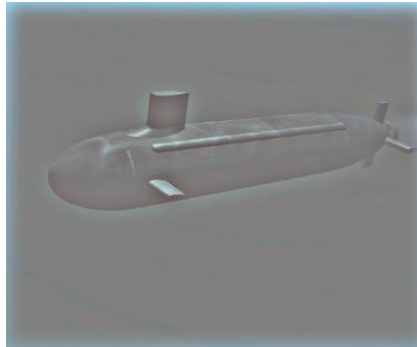
Original



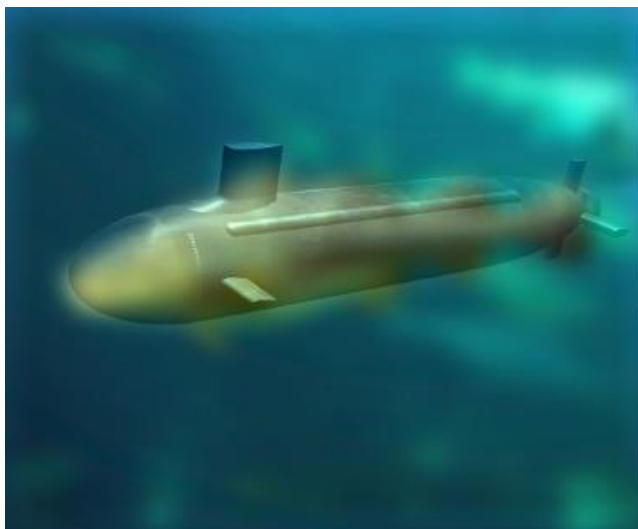
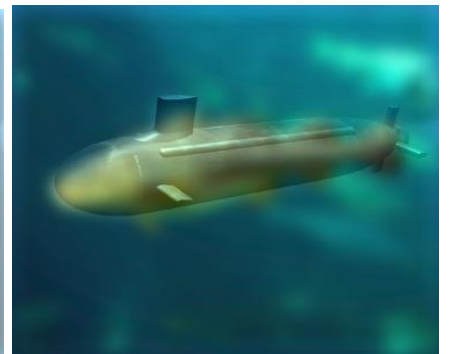
Low Freq



High Freq

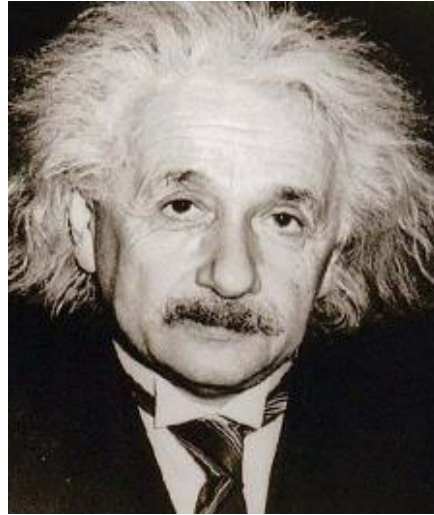


Hybrid



E. Marilyn (Low Freq) + Einstein (High Freq)

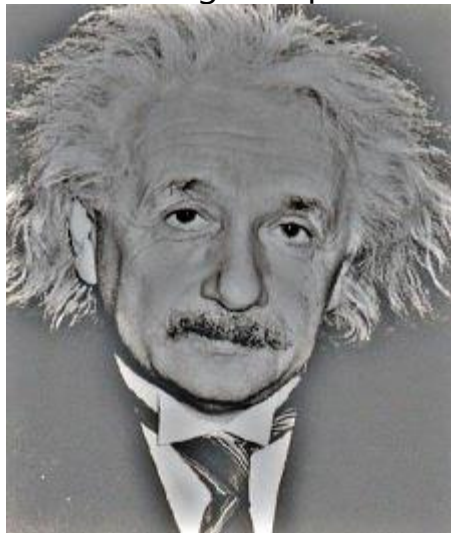
Original



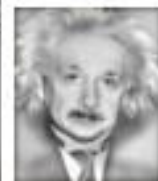
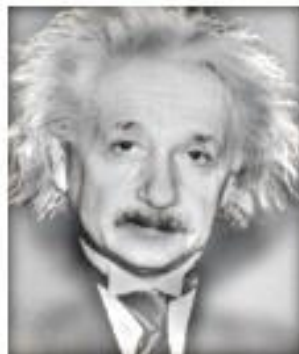
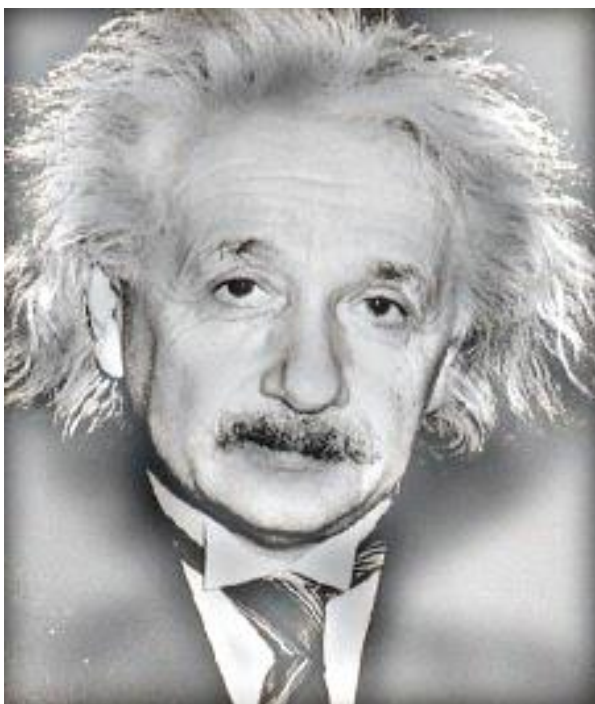
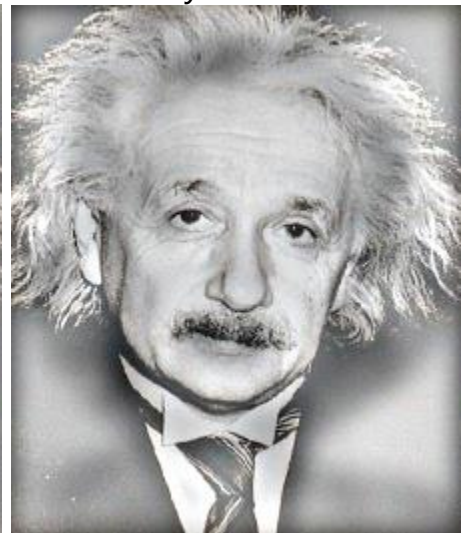
Low freq



High freq



Hybrid



● Discussion

1. why the high frequency images will be seen at close distance (larger one) while the low frequency images will be seen at far distance (smaller one)?

高頻的影像成分顯現出來的是影像的細節(ex.紋路、線條、亮暗、輪廓等)，而低頻的影像成分則是其餘的部分(例如：顏色分布、位置等)。舉例來說(以下圖為例)，從低頻影像我們能看得這張圖片中有一個很像鳥類的生物，圖片的背景與生物能區分出來，但我們沒辦法分出這是哪種鳥類因為他的細節特徵被消除了。



但是如果從下面這張圖片來看，我們就能很清楚的看得出這是哪種鳥類，我們甚至連他的羽毛的紋路、細節都能很清楚的看出來。這就是高頻的影像成分以及低頻的影像成分的區別。



至於為什麼在 Hybrid image 中我們在近距離看到的是高頻的部分而遠距離看到的是低頻的部分，我認為這就與我們的眼睛有關。眼睛也是有解析度的，在近距離觀看時，影像佔據大多數的視野範圍，影像的解析度與原先相距不大，因此人眼可以很清楚的看出影像中的細節(也就是高頻的成分)；相反的，當我們遠距離觀看時，影像只占視覺的一小部分，這時在我們視網膜上成巷的解析度就與原先影像的解析度低上許多(類似於將多個 pixel 合成一個 pixel 的感覺)，因此這時候的高頻部分就不明顯(被周圍 pixel compensate 掉)，低頻的影像更為 dominate，因此我們會看到低頻影像的部分。

2. Result of box filter

為了讓兩個 filter 的 size 相同，因此我挑 cutoff frequency = 7 \rightarrow Gaussian filter = $29 * 29$ ，box filter[i][j] = $1 / 29$ for all $i = 1 \sim 29, j = 1 \sim 29$ 。以下為兩者實做出來的成果

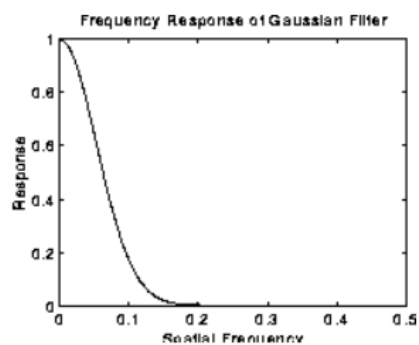
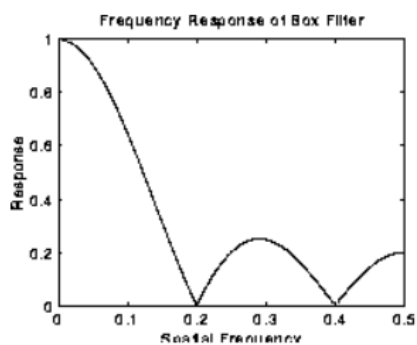
Gaussian



Box



從低頻的部分(第二排)比較能看出兩者的效果的差別，在 box filter 的 case 中，可以很明顯的在車輪中的紅色鐵條，而在 Gaussian filter 中紅色鐵條除了比較粗的那根以外幾乎都被刪除掉了，另外 Gaussian filter 的影像也比較模糊，我認為原因在於 Gaussian filter 的 decay 相較於 box filter 更為連續，且兩者的 frequency response 如下(左為 box filter、又為 gaussian filter)，box filter 在高頻部分仍然保有一小部分而 gaussian filter 則將高頻完全消除。

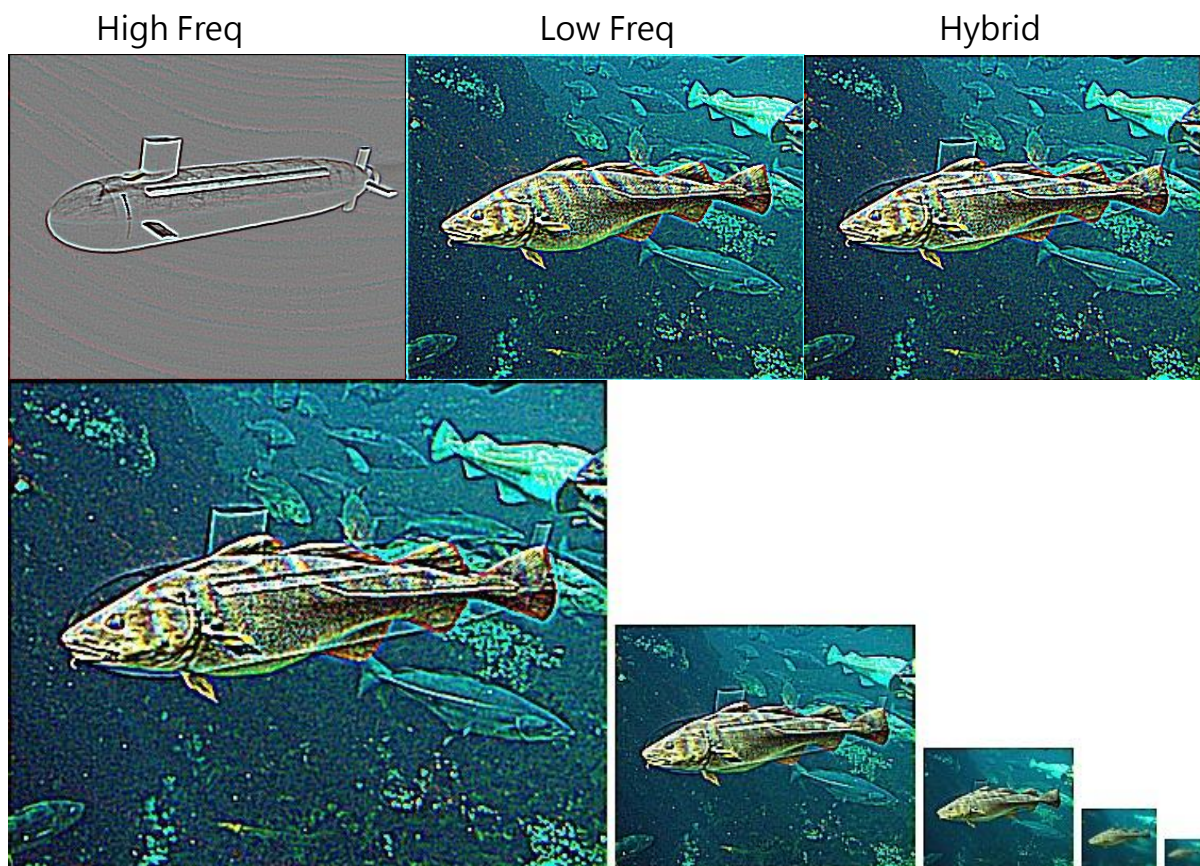


3. Result of reversing the flow

在前面我們是將影像經過 low pass 後與原先影像相減得到高頻影像成分，這部分我換一種做法，改採用 high pass filter，先將影像與他做 convolution 在與原先影像相減得到低頻影像成分。我採用的 high pass filter 如下，與助教提供的不同，原因在於我用助教提供的實作，但由於 filter kernel size，導致他擷取掉的頻率比較多，高頻成分不太明顯，所以我將 kernel size 放大一些($3 \times 3 \rightarrow 5 \times 5$)。

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & -8 & -1 & 1 \\ 1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

結果如下，與先前做出的結果相比，除了因為 kernel 大小不夠導致 hybrid 的效果不太明顯外，可以發現這種做法的顆粒感比較明顯，可以看到在低頻影像和高頻影像中都有明顯的噪聲。原先的方法做出來的圖片比較理想。



4. Result of different cutoff frequency

我嘗試 cutoff frequency = 3, 7, 11, 15 (由上到下)這四種，以下分別為他們的成果：



首先我先解釋一下 `fspecial` 這個 function 做了什麼，根據 MATLAB reference 上的敘述，`fspecial('Gaussian', hsize, sigma)`，中的 `hsize` 是 2D filter 的邊長、`sigma` 是這個 gaussian function 的 standard deviation，當 **standard deviation** 越大，他在頻域上的 **b** 的 **bandwidth** 就越小(如下圖所示)，能通過的頻率就越少。

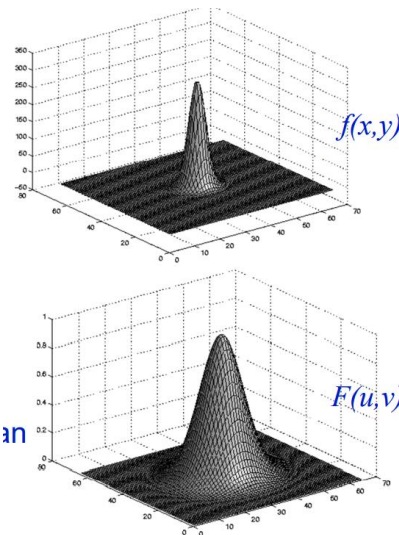
Gaussian centred on origin

$$f(r) = \frac{1}{2\pi\sigma^2} e^{-r^2/2\sigma^2}$$

where $r^2 = x^2 + y^2$.

$$F(u, v) = F(\rho) = e^{-2\pi^2\rho^2\sigma^2}$$

where $\rho^2 = u^2 + v^2$.



從結果上可以看出可以看的出來，當 **cutoff frequency** 越高時，低頻的成分越明顯；當 **cutoff frequency** 上升時，低頻影像逐漸變得模糊，原因在於 **cutoff frequency** 上升 → **filter size** 上升 → 越多點合成一個點 → 更模糊。另外，從 MATLAB 中的 tic toc 顯示的執行時間來看，當 **cutoff frequency** 上升時，因為 **filter kernel size** 變大，convolution 的計算量上升，導致程式執行的時間也跟著提升。

5. Alignment Problem

在製作 Hybrid image 時，我們會希望圖片的大小以及圖片中物體的大小以及位置是接近的。然而實務上我們所得到的圖片不一定有上述的特徵，例如以下兩張圖片，他們在圖片的大小、物體的大小甚至是物體的位置都不太一樣，所以我們需要做一些 **pre-processing**。



1689 * 1064



2000 * 2000

理想的方法是希望能夠透過例如 ML 或是特殊 filter 的方式找出物體的邊界並決定要如何將物體進行旋轉或平移讓兩個物體的位置重疊，但是這兩個方法我目前都做不到，所以我只能

用人工的方式去決定要旋轉以及平移多少去讓兩物體重疊。我的 alignment 步驟如下：

1. 先將兩張圖片縮放成相同的 size(可以運用前一個 Lab 的 resize 並搭配 MATLAB 實做裁剪功能。) → 轉成 1000*1000
2. 將物體進行旋轉以及平移讓兩物體重疊(可以運用前一個 Lab 做的 rotate)
3. 再次 resize 搭配裁剪將兩個物體的大小變成相同

實作的程式碼可以看 alignment.m，以下展示實做的成果：



可以發現只要經過一系列的預處理，我們就能透過我們的程式生出不錯的 Hybrid image。

要注意的是，我們在做 Hybrid image 時，要挑選形狀相似的物體，若是形狀(輪廓)相差很大的物體(例如球棒與棒球)這項做出來的效果就會比較差。

● Conclusion

這個 Lab 中我運用助教給的圖實作 Hybrid image，探討其在視覺上造成不同感覺的原因，並比較不同 filter 對影像造成的影響。我們必須挑選適當的 cutoff frequency、filter kernel size 才會有不錯的 Hybrid image 產生。另外，圖片本身也需要滿足一些條件，才會讓整個 Hybrid image 看起來比較合理，因此我們在實務上可能需要一些預處理，去解決實務上不同大小、物體位置不同的影像所形成的 Hybrid image 會沒有對齊的問題。

● References

- 教授與助教的講義
- [Spatial Filters - Gaussian Smoothing \(ed.ac.uk\)](#)
- [\[Python\]Gaussian Filter-概念與實作. 實作 Gaussian Filter | by 天道酬勤 | Medium](#)
- 2D image filter [Microsoft PowerPoint - lect2.pptx \(ox.ac.uk\)](#)