Computer Vision

HW1 Hybrid Images

學號:112061619

姓名:王證皓

1. Implementation

1.1 Image filtering

```
    # ============================ Start OF YOUR CODE ============================
    output = np.zeros_like(image)
    x=(imfilter.shape[0]-1)//2
    y=(imfilter.shape[1]-1)//2
    output_pad= np.zeros_like(image)
    output_pad = np.pad(image,((y,y),(x,x),(0,0)),'reflect')
    for h in range(image.shape[2]):
        for row in range(image.shape[0]):
            for col in range(image.shape[1]):
                output[row,col,h]=np.sum(output_pad[row:row+imfilter.shape[0],col:col+imfilter.shape[1],h]*imfilter)

    # ============================== END OF YOUR CODE ==============================
```

x 計算出 x 方向其中一側所要補的 pad 大小，y 則是 y 方向的 pad 大小，output_pad 代表的是利用 np.pad，將 image 補上 reflect 的 pad 後的圖像，接者使用三層的迴圈將影像與 filter 做乘積，並用 np.sum 加總以得到 output。

1.2 Extract and combine

```
    # ============================================================================
    # TODO: Use my_imfilter create 'low_frequencies' and
    # 'high_frequencies' and then combine them to create 'hybrid_image'
    # ============================================================================
    # ============================================================================
    # Remove the high frequencies from image1 by blurring it. The amount of
    # blur that works best will vary with different image pairs
    # ============================================================================
    low_frequencies = my_imfilter(image1, gaussian_filter)        # You need to modify here


    # ============================================================================
    # Remove the low frequencies from image2. The easiest way to do this is to
    # subtract a blurred version of image2 from the original version of image2.
    # This will give you an image centered at zero with negative values.
    # ============================================================================
    temp = my_imfilter(image2, gaussian_filter)         # You need to modify here
    high_frequencies=image2-temp


    # ============================================================================
    # Combine the high frequencies and low frequencies
    # ============================================================================
    hybrid_image = normalize(low_frequencies+high_frequencies)          # You need to modify here
```

此處使用 my_infilter 函數將圖像以 gaussian filter 移除高頻成分，得到 low frequencies，high frequencies 則是先移除高頻成分剩下低頻，再將原圖像減去低頻得到高頻，最後將兩者合併。

1.3 Others

(1)環境皆參照助教所給定的版本

```
conda create -n <env_name> python=3.8
conda activate <env_name>
conda install opencv=4.6.0
conda install scipy=1.10.1
conda install matplotlib=3.7.2
```

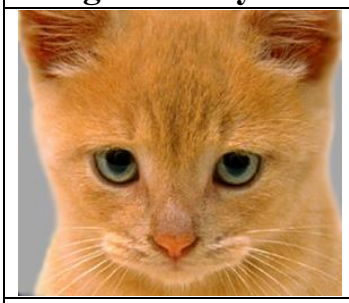
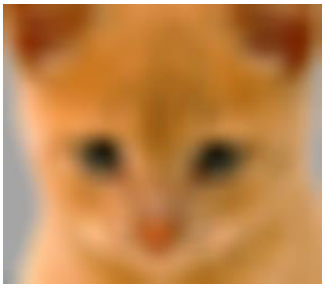
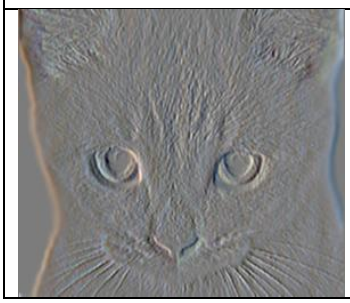
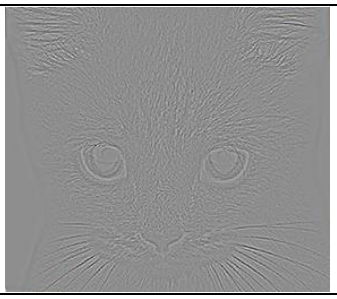
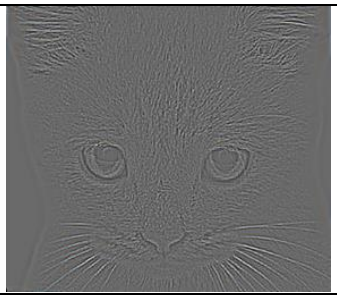
(2)以下指令分別為執行 filter 的測試檔以及 hybrid 的圖像

```
python .\code\hw1_test_filtering.py
python .\code\hw1.py
```

2. Experiments

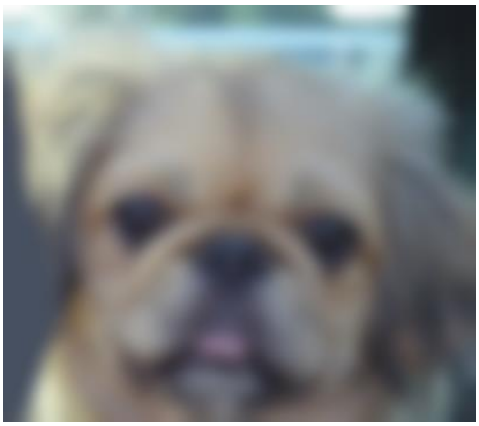
2.1 Hybrid Image

(1)filter

Image/Identity	Box filter	Gauss filter
		
Sobel filter	**Laplacian filter**	**High pass filter**
		

從以上圖片中，可看到 Box filter 移除了 sharp feature，Gauss filter 移除了高頻成分，Sobel filter 留下了 vertical edge。

(2)hybrid

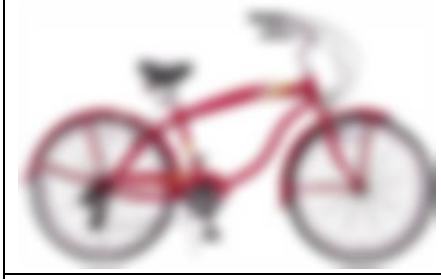
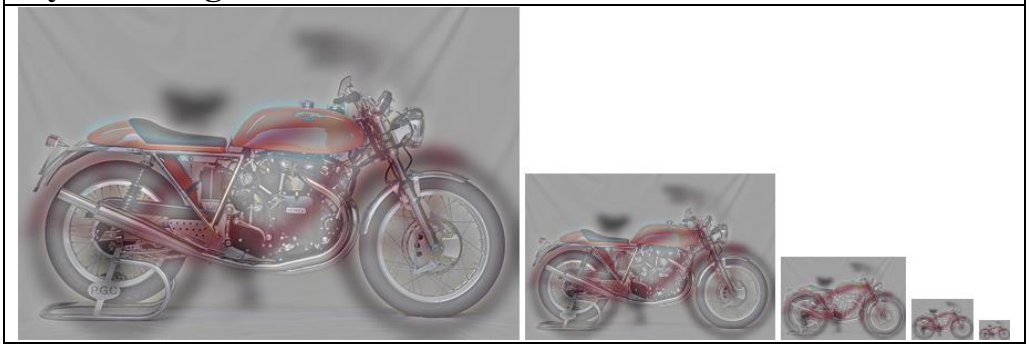
Low frequencies	High frequencies
	
Hybrid image	
	

低頻的狗圖像會變得較 smooth 以及模糊化，高頻的貓圖像則會輪廓會更加明顯，混和後的圖像則是距離越近的話，高頻成分較明顯，越像貓，越遠的話，低頻成分較明顯，則越像狗。

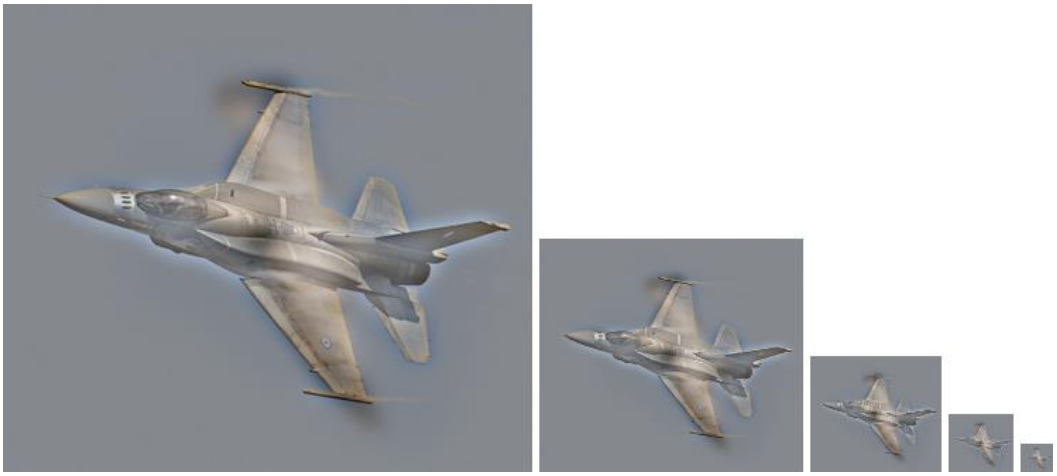
2.2 Other hybrid images

以下分別合併了四種成對的圖像，且只有在圖像的尺寸有對齊以及形狀對稱下，圖像才能完美的混和。

(1)bicycle+motorcycle

Low frequencies	High frequencies
	
Hybrid image	
	

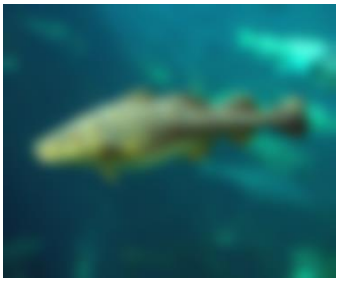
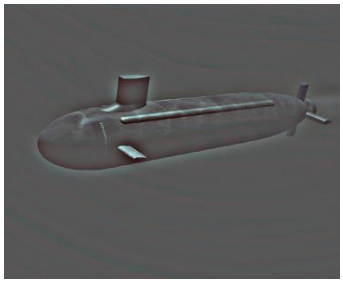
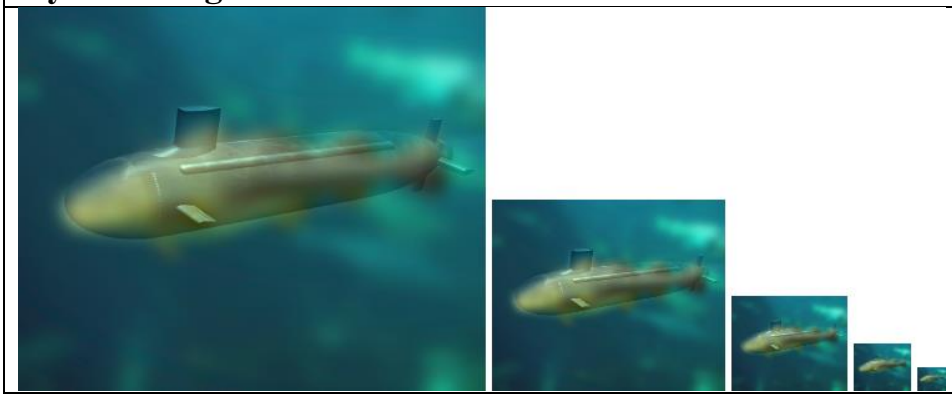
(2)bird+plane

Low frequencies	High frequencies
	
Hybrid image	
	

(3)Einstein+marilyn

Low frequencies	High frequencies
	
Hybrid image	
	

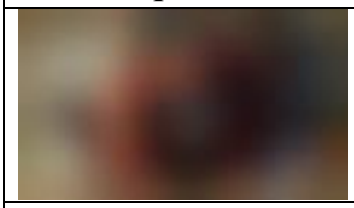
(4)fish+submarine

Low frequencies	High frequencies
	
Hybrid image	
	

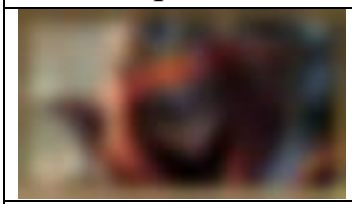
2.3 Customized hybrid images

以下使用了一款名叫英雄聯盟的遊戲的其中兩個腳色，可以看到如使用 cutoff frequency=7 的情況下，圖像會極度的模糊，但在 cutoff frequency=3 的情況下，就能很明顯的辨認出各個腳色，由此可知 cutoff frequency 的重要性。

(1)cutoff_frequency=7

Low frequencies	**High frequencies**
	
Hybrid image	
	

(2)cutoff_frequency=3

Low frequencies	**High frequencies**
	
Hybrid image	
	

3. Discussion

我發現即使是兩張貓與狗的圖片與前者相同，但當交換順序後，所得到的影像也會有很大的差異，應用的部分，我認為可以應用在高速公路要下交流道的速限指示牌上，遠距離還在高速公路上時，呈現出較高的速限，但當近距離要下交流道時，變呈現較低的速限，如此一來，能只用一張指示牌來標示以減少成本。

Low frequencies	**High frequencies**
	
Hybrid image