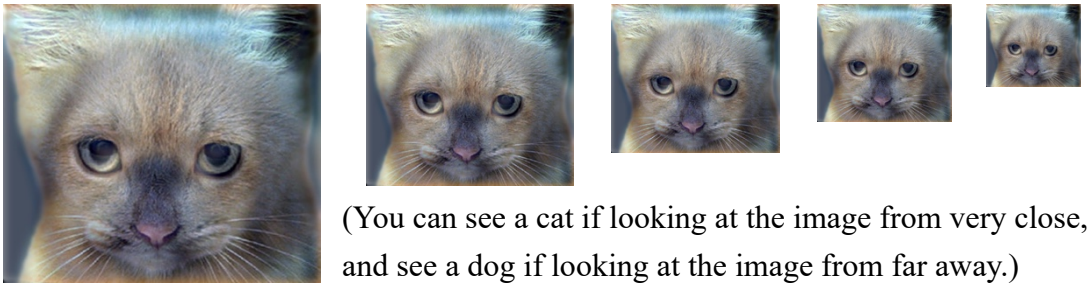
# Homework #1 Hybrid Images

*Assigned on September 14, 2022*
*Due by September 28, 2022*
111061553 謝霖泳

## Overview



(You can see a cat if looking at the image from very close, and see a dog if looking at the image from far away.)
The goal of this assignment is to write an image filtering function and use it to create hybrid images using a simplified version of the SIGGRAPH 2006 paper (slides) by Oliva, Torralba, and Schyns.

*Hybrid images* are static images that present different interpretation as the viewing distance changes. The basic idea is that high-frequency signal (e.g., edges, textures, etc.) tends to dominate perception when closely observing an object. However, from a distance, only the low-frequency (smooth) part of the signal can be seen. By blending the high-frequency portion of one image with the low-frequency portion of another, you get a hybrid image that leads to different interpretations at different distances.

## Details

Please refer to README.md in the homework folder to see more details about how to finish your code implementation in this homework.

## 1 Implementation (50%)

### 1.1 Image filtering (20%)

Please finish the function **my_imfilter** in the file **my_imfilter.py** and briefly describe your implementation ideas.

先利用 np.pad()對原始 image 的 3 個 channel 的長及寬進行 zero padding，接著利用兩層 for loop 對所有的位置進行 filter 點對點相乘運算，最後 return output 即為所求。

```
    output = np.zeros_like(image)
    v_offset = (imfilter.shape[0] - 1 ) // 2
    h_offset = (imfilter.shape[1] - 1 ) // 2
    ch0 = np.pad(image[:, :, 0], (v_offset, h_offset), mode='constant')
    ch1 = np.pad(image[:, :, 1], (v_offset, h_offset), mode='constant')
    ch2 = np.pad(image[:, :, 2], (v_offset, h_offset), mode='constant')
    for r in range(image.shape[0]):
        for c in range(image.shape[1]):
            output[r][c][0] = np.sum(imfilter * ch0[r: r + imfilter.shape[0],
c: c + imfilter.shape[1]])
            output[r][c][1] = np.sum(imfilter * ch1[r: r + imfilter.shape[0],
c: c + imfilter.shape[1]])
            output[r][c][2] = np.sum(imfilter * ch2[r: r + imfilter.shape[0],
c: c + imfilter.shape[1]])
    return output
```

### 1.2 Extract and combine the high-frequency and low-frequency signals (20%)

Please finish the **TODO** in the file **hw1.py**.

image1 經過 Gaussian filter 運算完之後就是 image1 的低頻部分，再將 image2 經過上述相同運算後得到 image2 的低頻部分，將原始的 image2 原圖減去該圖低頻部分，即可得到 image2 的高頻部分。最後，將 image1 的 low freq.加上 image2 的 high freq.相加做 normalize 之後即可得到 hybrid image。

```
    low_frequencies = my_imfilter(image1, gaussian_filter)
    high_frequencies = image2 - my_imfilter(image2, gaussian_filter)
    hybrid_image = normalize(low_frequencies + high_frequencies)
```

### 1.3 Others (10%)

Please list the additional packages and versions required in your implementation and describe how to run your code. (make sure we can run your code)

(1) Additional packages

- python = 3.8.13
- opencv = 4.5.5

(2) How to run my code

```
$ python hw1.py
```

# 2 Experiments (40%)

## 2.1 Hybrid Image (15%)

Put your hybrid result from the cat-dog pair and briefly explain your result.

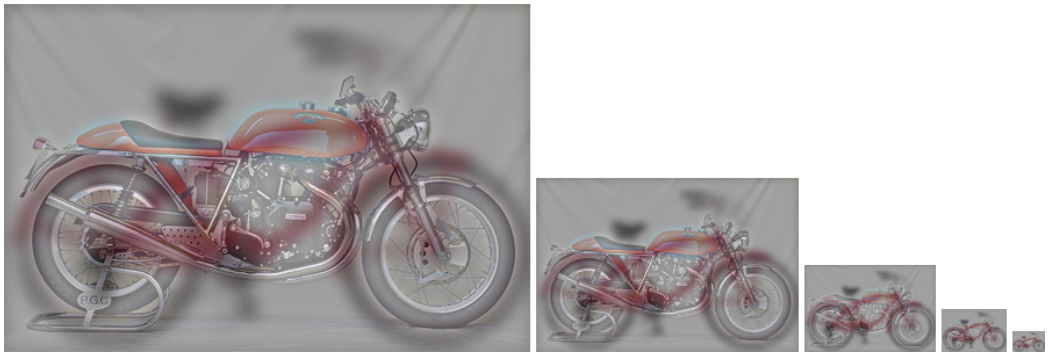
狗是 low frequency，貓是 high frequency，並將 cutoff frequency 設為 7，即可得到上面的 hybrid image。其中，遠看會看到 low frequency 的部分，也就是一隻狗，而近看會看到 high frequency 的部分，即一隻貓。

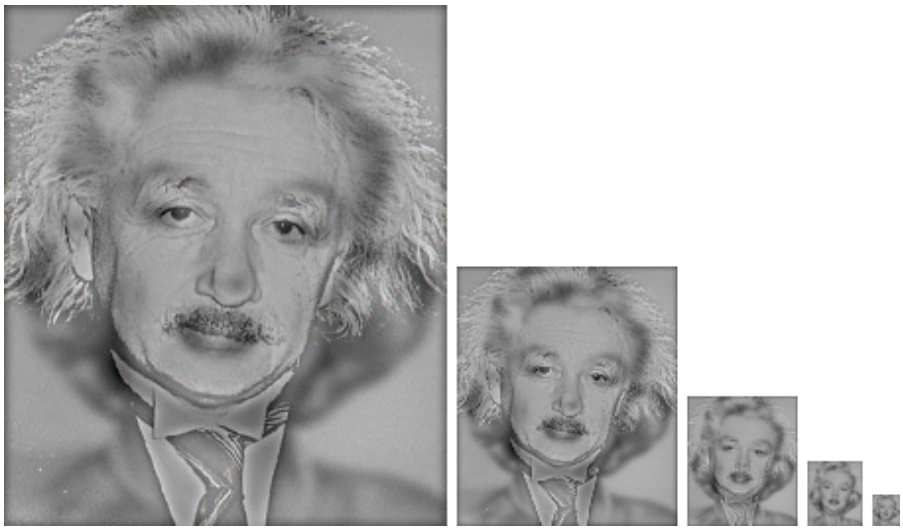
## 2.2 Other hybrid images (15%)

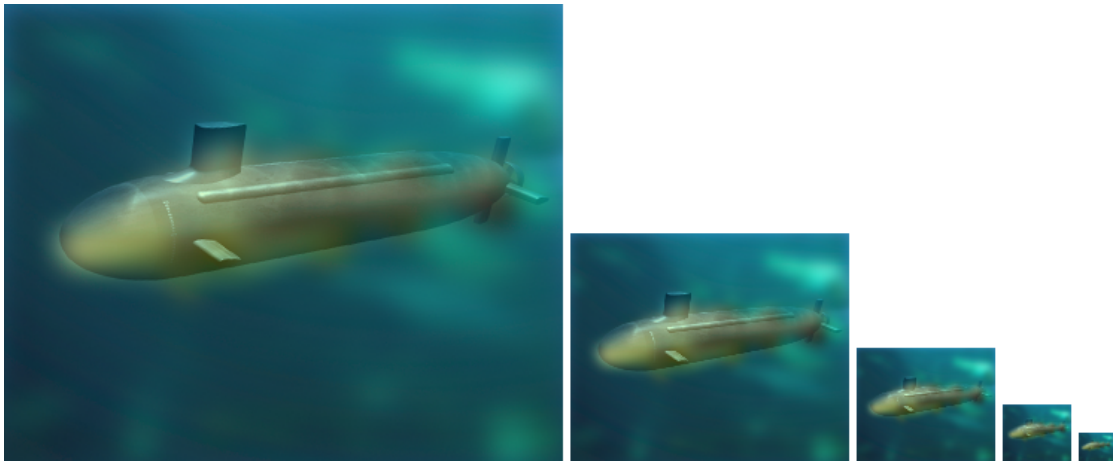
Try different pairs of pictures in the folder **/data** and put your results here. Comparing the result of Problem 2.1, what's the difference?

下一頁將展示四組 hybrid image 的實作結果，分別為腳踏車與機車、鳥與噴射機、Einstein 與 Marilyn、魚與潛水艇。在這四個例子中，都可以發現遠看和近看會看到兩個不同的東西，也就是低頻部分以及高頻部分。

但是每張圖片在實作時還有一個些微的不同，就是 cutoff frequency，在 template code 之中，cutoff frequency 的值被預設為 7，但在這幾個例子中會發現有些 hybrid image 如果 cutoff frequency 設為 7 的效果並不好，遠看或近看都是看到同一個東西，也就是高頻或低頻某一邊的佔比太高，導致無法達成 hybrid image 的效果。因此，如果發現 hybrid image 做出來的效果不佳時，我會調整 cutoff frequency 的數值，使得 hybrid 的效果能夠變得比較明顯。



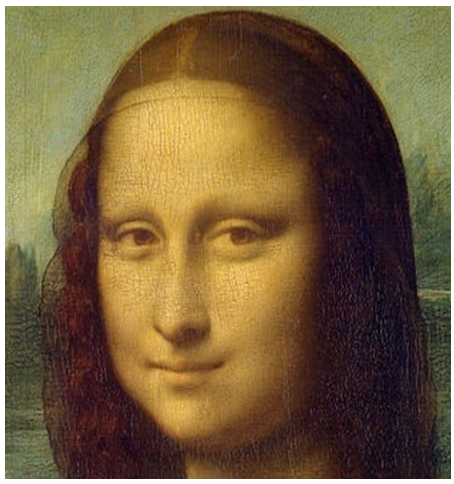




## 2.3 Customized hybrid images (10%)

Gather your own picture pairs and show your results of hybrid results.

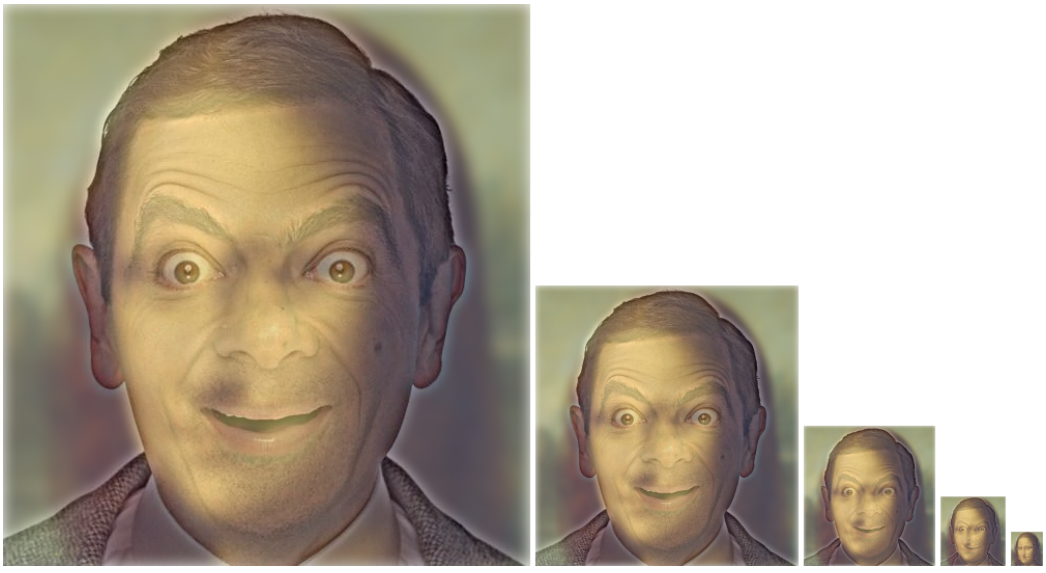
(1) Original picture pairs

(2) High / Low frequencies

| High frequency | Low frequency |
|---|---|
|  |  |

(3) Hybrid image



## 3 Discussion (10%)

Do you discover anything special in your experimental results?
What applications do you think this technology can be used for?

在幾個 hybrid image 的實驗中，我發現作業提供的四組 hybrid image pair，有些都算是 hybrid image 的經典範例，比如愛因斯坦的例子。因此，這幾個例子中兩張 image 的相對位置都是相似的，像是噴射機的機翼對到鳥的翅膀、腳踏車和機車的輪子位置也是一致的等等。

但是如果要自己找 image pair 來實作的話，我認為要找到適合的例子並不容易。如果兩張照片的內容物相對位置差太多的話，就會造成最後做出來的效果不好。因此，我最後找了兩張知名的人臉，做出的 hybrid image 效果看起來還不錯，但是因為五官位置沒有剛好完全 align 的關係，仔細看還是可以發現一點怪怪的。

我認為 hybrid image 可以應用在廣告行銷的手法，只要商家能做出一張夠有梗的 hybrid image，內容又和自己在賣的產品相關的話，在網路上就會造成一定的知名度，便可達到打廣告的目的。

## 4 Requirement

You should package the required files in a folder named **HW1_{studentID}**.

1. ***.py in in folder **/code** (all required files to run your code)
2. ***.png in folder **/results** (your generated hybrid images)
3. Your report with filename **{studentID}_report.pdf**

Compress the entire folder **HW1_{studentID}** into **HW1_{studentID}.zip** and submit it to e-learning.

Any wrong format or file arrangement will get 5% punishment each time.

## 5 Appendix and Reference

[1] Assignment modified by Min Sun based on James Hays and Derek Hoiem's previous developed projects.

[2] Oliva, Aude, Antonio Torralba, and Philippe G. Schyns. "Hybrid images." ACM Transactions on Graphics (TOG) 25.3 (2006): 527-532.