Homework #1 Hybrid Images

Assigned on September 14, 2022
Due by September 28, 2022

Overview

(You can see a cat if looking at the image from very close, and see a dog if looking at the picture 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.

```
# Image padding
image = np.pad(image, ((imfilter.shape[0]//2, imfilter.shape[0]//2),
              (imfilter.shape[1]//2, imfilter.shape[1]//2), (0, 0)), 'constant', constant_values=0)
```

In image padding, I referenced from the np.pad document and used this function.
The image has expanded the size of filter mod 2 pixels.

```
# Image filtering with numpy.sum
for k in range(image.shape[2]):
    for i in range(image.shape[0]-imfilter.shape[0]+1):
        for j in range(image.shape[1]-imfilter.shape[1]+1):
            output[i, j, k] = np.sum(
                image[i:i+imfilter.shape[0], j:j+imfilter.shape[1], k] * imfilter)
```

In image filtering, I use the triple loop for calculation.
The first loop is for the dimension of the channel.
The second loop is for the dimension of the height.
The third loop is for the dimension of the width.

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

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

```
 1  # =====================================================================
 2  # TODO: Use my_imfilter create 'low_frequencies' and
 3  # 'high_frequencies' and then combine them to create 'hybrid_image'
 4  # =====================================================================
 5  # =====================================================================
 6  # Remove the high frequencies from image1 by blurring it. The amount of
 7  # blur that works best will vary with different image pairs
 8  # =====================================================================
 9  # You need to modify here
10  low_frequencies = my_imfilter(image1, gaussian_filter)
11
12  # =====================================================================
13  # Remove the low frequencies from image2. The easiest way to do this is to
14  # subtract a blurred version of image2 from the original version of image2.
15  # This will give you an image centered at zero with negative values.
16  # =====================================================================
17  # You need to modify here
18  high_frequencies = image2 - my_imfilter(image2, gaussian_filter)
19
20  # =====================================================================
21  # Combine the high frequencies and low frequencies
22  # =====================================================================
23  # You need to modify here
24  hybrid_image = low_frequencies + high_frequencies
25  hybrid_image = normalize(hybrid_image)
```

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)
No additional packages.
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.

2.2   Other hybrid images (15%)

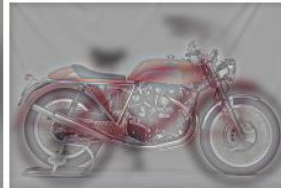Try different pairs of pictures in the folder **/data** and put your results here.

Comparing the result of Problem 2.1, what's the difference?

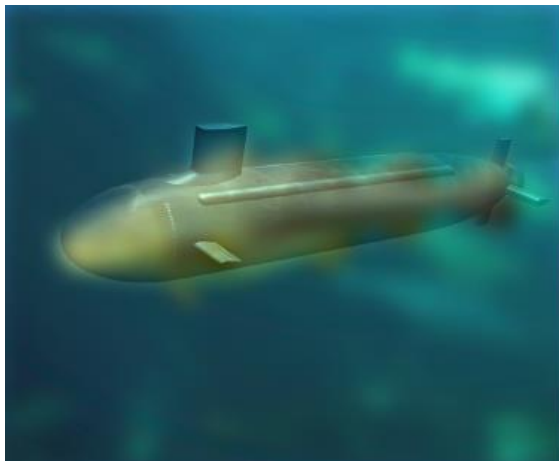**I think the human face is not as apparent as a cat or dog.**

Einstein and Marilyn:

Bicycle and Motorcycle:

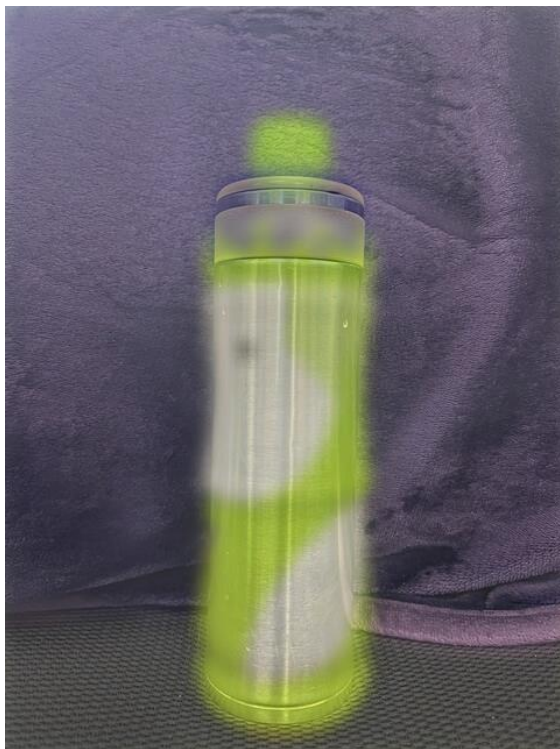**Fish and Submarine:**

Bird and Plane:

2.3   Customized hybrid images (10%)

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

3   Discussion (10%)

Do you discover anything special in your experimental results?

1. If image1 is more colorful than the image2, the colorful one is more obvious than another in a hybrid image.
2. I think the human face pairs are not apparent as the other animals

What applications do you think this technology can be used for?

1. Runway show
2. Toy

4   Requirement

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

1.   ***.py 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.