Homework 2 Writeup

Instructions

- Describe any interesting decisions you made to write your algorithm.
- Show and discuss the results of your algorithm.
- Feel free to include code snippets, images, and equations.
- Use as many pages as you need, but err on the short side If you feel you only need to write a short amount to meet the brief, th
- **Please make this document anonymous.**

Overview

In this homework, I wrote code to create hybrid images. First, image processing using convolution is written in myfilter2D function. Therefore, depending on the image and filter inputs, the output image suitable for the purpose can be obtained. Second, after selecting two images, through myfilter2d function, one image gets the low frequency part and the other image gets the high frequency part. A hybrid image is created by combining each of the obtained images.

Details in Code: my filter 2D

I will explain the code written in myfilter2D function in detail.

First, the case where the width and height of the filter are even is excluded.

```
if (ker_height % 2 == 0) or (ker_width % 2 == 0):
  print("Filter length is even")
  return
```

Then, when the filter is an identity filter, the image is exported as it is.

```
identity_kernel = np.zeros(kernel.shape)
identity_kernel[int((ker_height - 1) / 2), int((ker_width - 1)
   / 2)] = 1
if np.all(kernel == identity_kernel):
  return image
```

Next, reflect padding and zero padding are distinguished, and color image and grayscale image are distinguished through if-else syntax. And padding is done so that the output image has the same size as the input image.

```
if reflect == False:
  if image.ndim == 3:
    padded_image = np.pad(image,((pad_height,
        pad_height),(pad_width, pad_width),(0, 0)),'constant')
  else:
    padded_image = np.pad(image,((pad_height,
        pad_height),(pad_width, pad_width)),'constant')
else:
  if image.ndim == 3:
    padded_image = np.pad(image,((pad_height,
        pad_height),(pad_width, pad_width),(0, 0)),'reflect')
  else:
    padded_image = np.pad(image,((pad_height,
        pad_height),(pad_width, pad_width)),'reflect')
```

Next, the filter is rotated 180 degrees for convenience of calculation.

```
rotated_kernel = cv2.rotate(kernel,cv2.ROTATE_180)
```

The following shows the correlation (because the filter was rotated 180 degrees). At this time, color image and grayscale image are distinguished.

```
if image.ndim == 3:
  for i in range(result_image.shape[0]):
    for j in range(result_image.shape[1]):
      result_image[i,j,0] =
          np.sum(padded_image[i:i+ker_height,j:j+ker_width, 0]
          * rotated_kernel)
      result_image[i,j,1] =
          np.sum(padded_image[i:i+ker_height,j:j+ker_width, 1]
          * rotated_kernel)
      result_image[i,j,2] =
          np.sum(padded_image[i:i+ker_height,j:j+ker_width, 2]
          * rotated_kernel)
else:
  for i in range(result_image.shape[0]):
    for j in range(result_image.shape[1]):
      result_image[i,j] =
          np.sum(padded_image[i:i+ker_height,j:j+ker_width] *
          rotated_kernel)
```

Details in Code: generate hybrid image

The image with low frequency is obtained using Gaussian filter.

```
kernel = cv2.getGaussianKernel(cutoff_frequency * 4 + 1,
    cutoff_frequency)
kernel = np.matmul(kernel, kernel.T)
low_frequencies = my_filter2D(image1, kernel)
```

The high frequency image is obtained by removing the low frequency part from the original image.

```
high_frequencies = image2 - my_filter2D(image2, kernel)
```

The low-frequency and high-frequency images are combined. At this time, the pixel values are adjusted so that they do not exceed values between 0 and 1.

```
hybrid_image = low_frequencies + high_frequencies
hybrid_image[hybrid_image > 1] = 1
hybrid_image[hybrid_image < 0] = 0
```

Result: myfilter2D Test

1. Identity image
   This is the same as the original image.

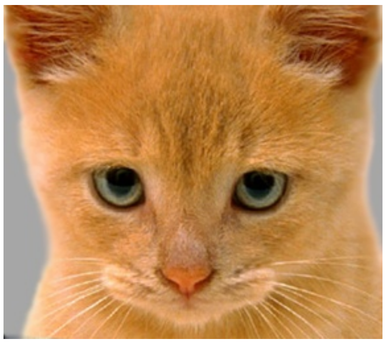

Figure 1: Identity image

2. Low-pass image
   The image on the left came out slightly blurry and the image on the right came out much blurry.

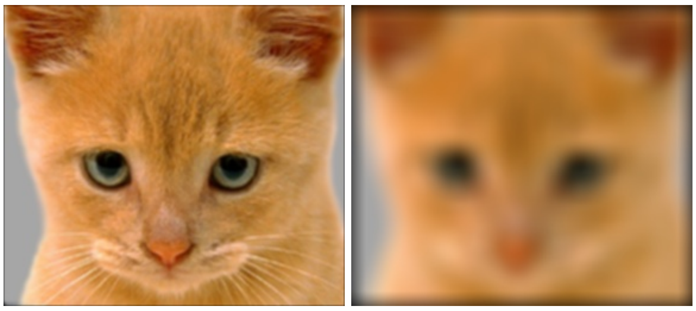

Figure 2: *Left:* Blur image *Right:* large Blur image

3. Sobel image
   We can get vertical edge.

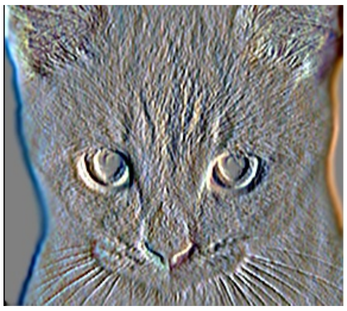

Figure 3: Solbel image

4. High-pass image
   We can clearly see the lines in the cat image.

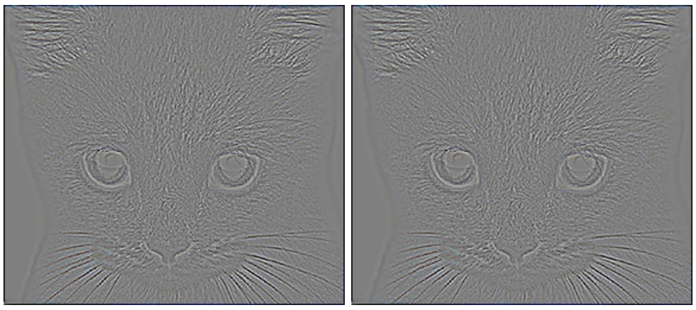

Figure 4: *Left:* Laplacian image *Right:* High pass image

5. Zero padding vs. Reflect padding
   The zero padding shows black areas at the corners. However, the reflective padding appears naturally.

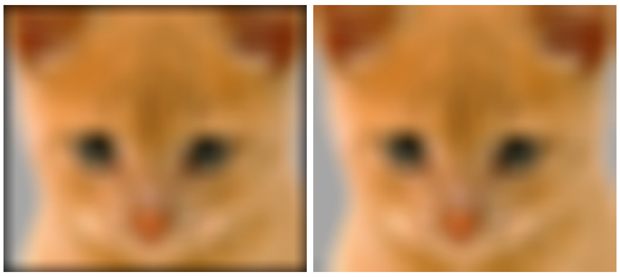
Figure 5: *Left:* Zero padding *Right:* Reflect padding

Result: Hybrid Image

1. Cat and Dog
   The cat convolved with a high frequency filter has clear lines, and the dog with a low frequency filter is blurry.

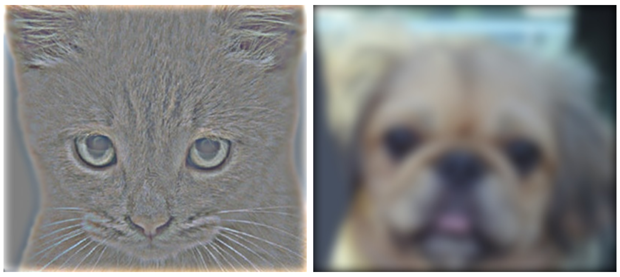
Figure 6: *Left:* High frequencies *Right:* Low frequencies

The large image looks like a cat, but the small image looks like a dog.

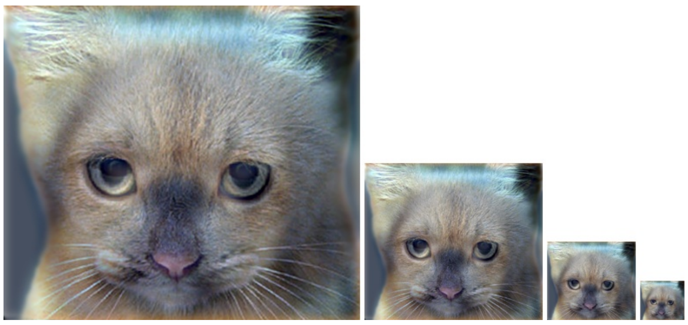
Figure 7: Cat and dog hybrid images

2. Einstein and Marilyn
   Einstein convolved with a high frequency filter has clear lines, and Marilyn with a low frequency filter is blurry.

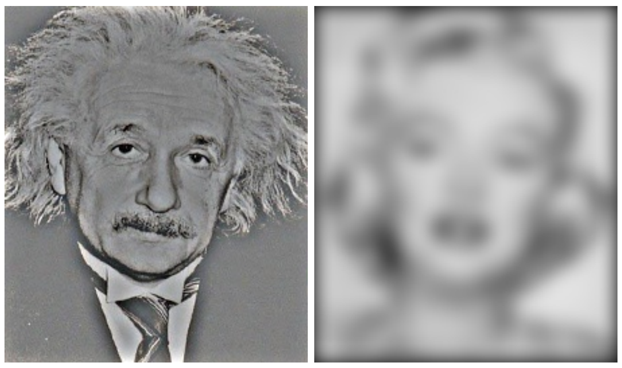

Figure 8: *Left:* High frequencies *Right:* Low frequencies

The large image looks like Einstein, but the small image looks like Marilyn.

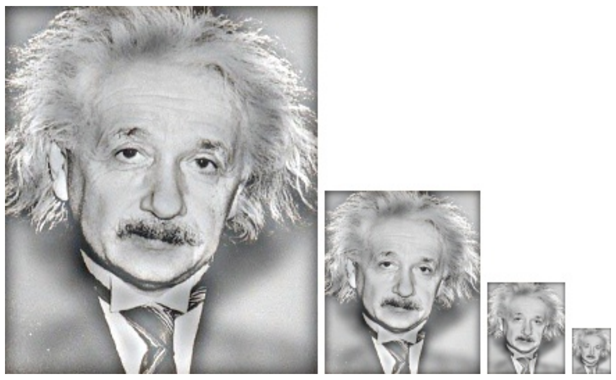

Figure 9: Einstein and Marilyn hybrid images

3. Bicycle and Motorcycle
   The bicycle convolved with a high frequency filter has clear lines, and the motorcycle with a low frequency filter is blurry.

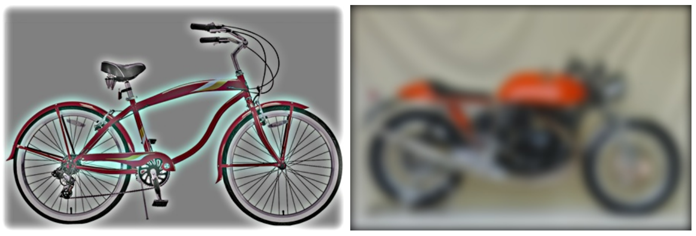

Figure 10: *Left:* High frequencies *Right:* Low frequencies

The large image looks like the bicycle, but the small image looks like motorcycle.

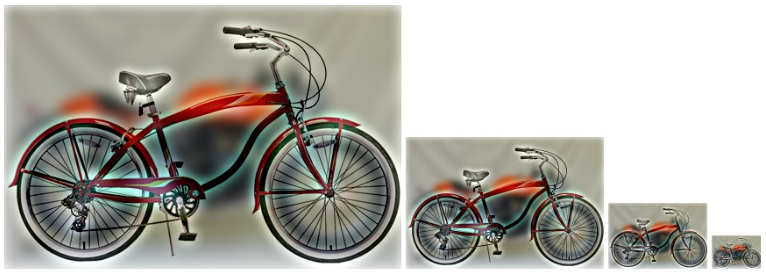

Figure 11: Bicycle and Motorcycle hybrid images

4. Bird and Plane
   The bird convolved with a high frequency filter has clear lines, and the plane with a low frequency filter is blurry.

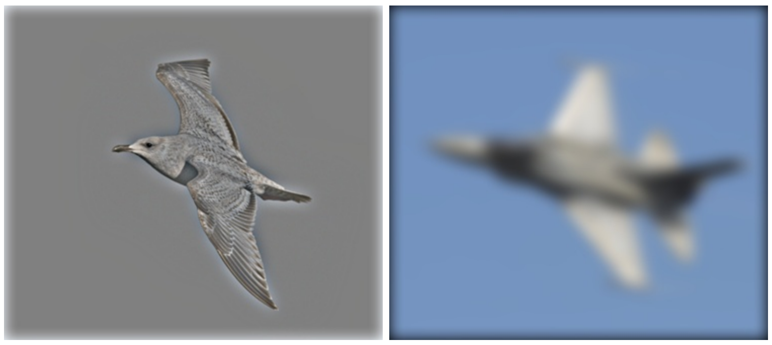

Figure 12: *Left:* High frequencies *Right:* Low frequencies

The large image looks like the bird, but the small image looks like plane.

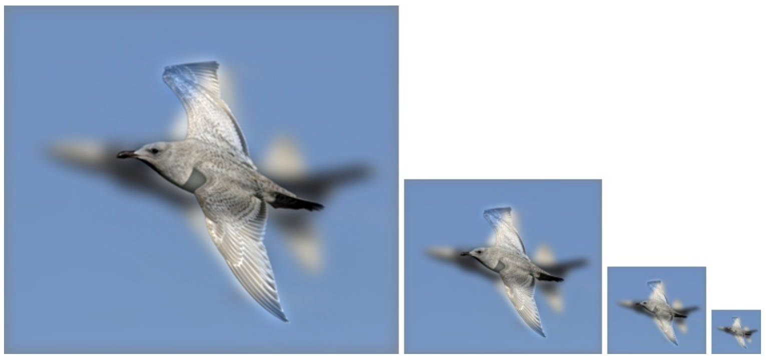

Figure 13: Bird and Plane hybrid images

5. Fish and Submarine
   The fish convolved with a high frequency filter has clear lines, and the submarine with a low frequency filter is blurry.

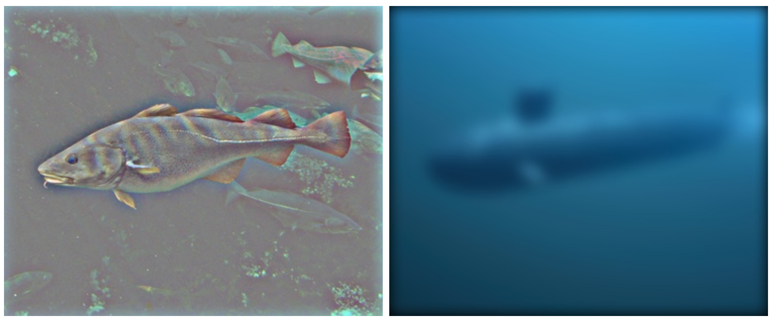

Figure 14: *Left:* High frequencies *Right:* Low frequencies

The large image looks like the fish, but the small image looks like submarine.

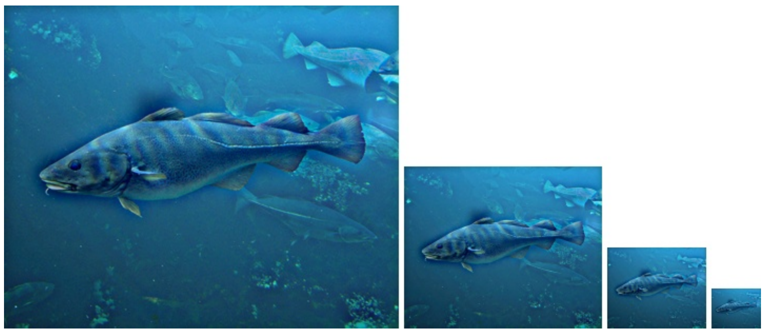

Figure 15: Fish and Submarine hybrid images

Summary

In the image, frequency indicates how large the change in pixel values is. By performing the Fourier transform on the image, the amplitude and phase according to the frequency can be obtained. At this time, if we get the low frequency part, we can get the blurred image, and if we get the high frequency part, we can see the edge of the image. From the two images, I obtained the low-frequency part and the high-frequency part, respectively, and combined them to obtain the desired hybrid image. If the size of the image is large, the edges are clearly visible, so we can see the high frequency image. If the size of the image is small, the edges are not visible and the image is viewed as a whole, so we can see the low frequency image.