Lab 2: Image filtering and hybrid images

Cano Andrés, Lorenzo
736078
Tierno Jiménez, Carlos
705548

March 18, 2021

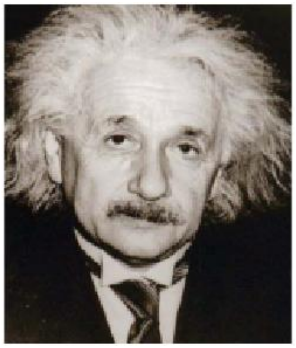
1 Image Filtering

Image filtering (or convolution) can be achieved with kernel convolutions working over the raw image or working with signals multiplications working on the Fourier domain. We are using the Fourier domain in this assignment, to take advantage of the high optimization that Fast Fourier Transformation algorithms bring over kernel convolutions.

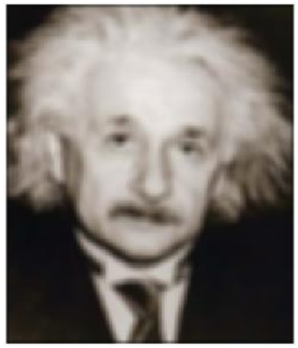
The function $my\_imfilter$ works over all the scenarios required at this part of the laboratory. Being reached by implementing the operations in the following way:

1. **Grayscale/Color support:** Using matrix operations, Matlab deals automatically with the third matrix dimension.
2. **Arbitrary shaped filters:** Padding the dimensions of the image and the kernel so their dimensions equal sum of their original sizes in order to be able to apply the kernel successfully.
3. **Pad the input image with zeros:** The fft2 function automatically fills the augmented dimensions with zeros.
4. **Return a filtered same size image:** fft2 pads filling with zeros; adding the same amount of them all over the matrix. Therefor, the matrix has been cropped by half of the size of the kernel at the bottom, top, left and right sides of the image.

In order to test the function, *test_filtering.m* has been completed and executed. Giving the following results for each kernel type:

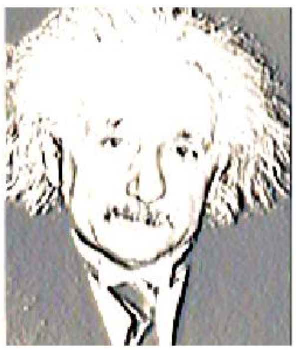


(a) Identity:
[0,0,0; 0,1,0; 0,0,0];

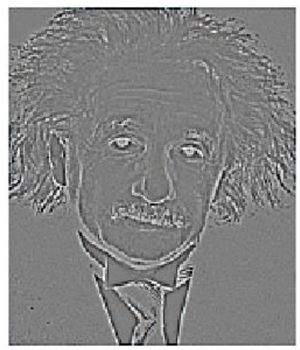


(b) Small Blur:
ones(5,5)

(c) Large Blur:
fspecial('gaussian', [31 31], 4)



(d) Sobel Op:
[1,0,-1;2,0,-1;1,0,-1]



(e) High-Pass Filter:
img - small_blur

2 Hybrid Images

Brief explanation: After loading the two images from which the hybrid image will be obtained:

1. The already implemented *align_image.m* function is called to align the two images.
2. The alignment leaves some very noticeable black margins. To avoid having them in the final image, a subsection of the aligned images is taken.
3. A Gaussian blur is applied to both images to get a low-frequency version of them.
4. For one of the two images, it's corresponding low-frequency version is subtracted. This way, the high-frequency version is obtained.
5. Finally, the high-frequency version of one image is summed to the low-frequency version of the other, obtaining the hybridized image.

Parameters: Depending on the pair of images used, the high frequency image can be barely noticeable in the hybrid image. In this cases we have found that the result can be greatly improved by multiplying the high frequency image by some value.

3 Questions

- Explicitly describe image filtering (the input, the transformation, and the output) and why it is useful for computer vision.
  - Image filtering takes as input 2 matrices. One, the image to be filtered, and the other, the kernel, that specifies the kind of filtering. The process itself is the convolution of the kernel over the image, and the result is the corresponding convoked image.
  - The usefulness for computer vision is related to the great variety of kernels that can be used. As different kernels allow to reduce noise, blur the image, detect borders and more.
- What is the difference between a high pass filter and a low pass filter in how they are constructed, and what they do to the image?
  - Low pass filter are usually constructed so that the resulting convolution for a certain pixel results in a weighted average of the surrounded pixels, but not giving any particular importance to any direction. On the other hand, high pass filter are not "isotropic", they tend to weight different directions differently, using negative values also. So that if a value increases in that certain direction around the pixel, this gradient is enhanced, and not supressed.
  - Their impact on images differ in the kind of features they enhance and/or decrease. While high pass filters enhance smaller features (edges, points, noise, anything that causes a sharp variation in the intensity of the pixels), the low pass filters suppress those smaller features, and so, the resulting image is more blurry.
- How does the Fourier transform relate to image filtering?
  - Image filtering is a 2D convolution. And, as a convolution in spatial domain is a multiplication in the frequency domain, one could apply filtering to an image by multiplying the Fourier transforms of the kernel and the image, and applying the inverse Fourier transform to the result.

3.1 Hybrid Images Results

At this section, the different combinations of images are exposed.
To reach this results, the low and high frequencies has been pondered based on the mean value of the mean color value of each pair. This is done with the following code:

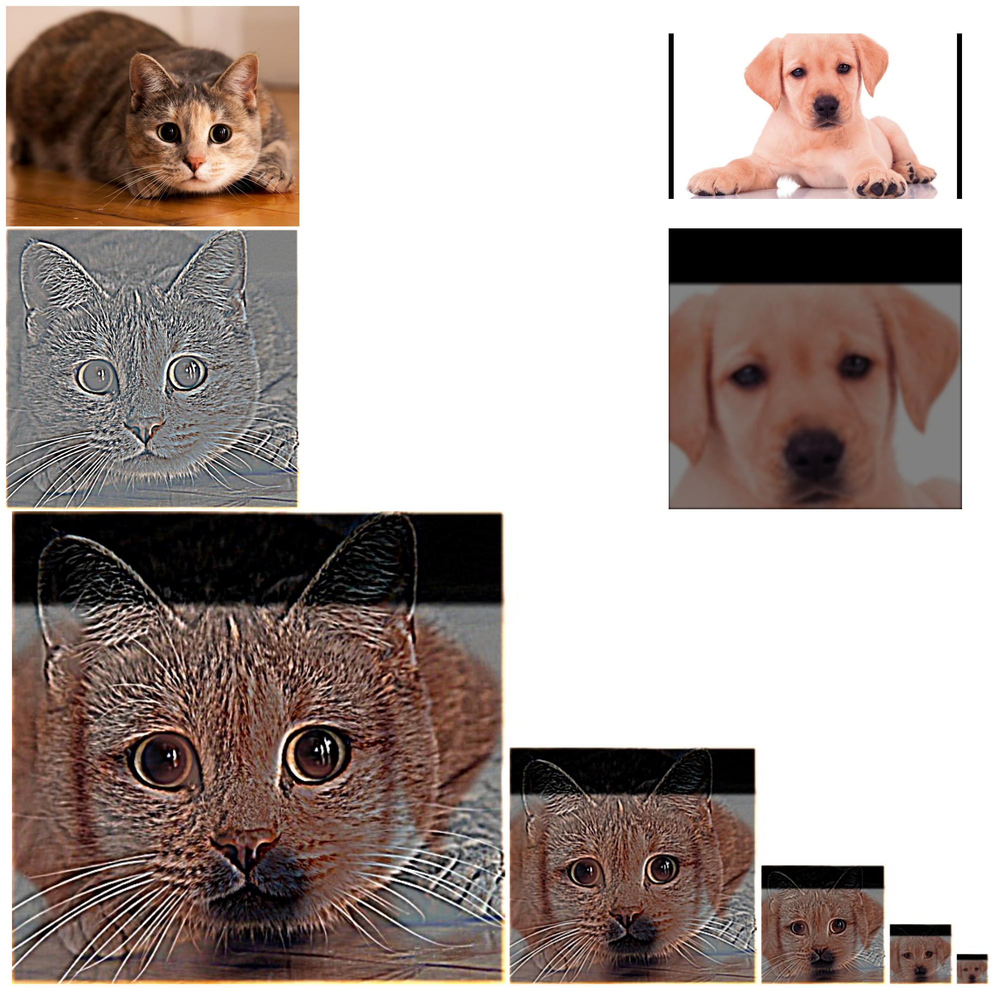
```
mean_lf = mean(low_frequencies, 'all');
mean_hf = mean(high_frequencies, 'all');
tmean = (mean_lf+mean_hf)/2;

low_frequencies = low_frequencies*tmean/mean_lf;
high_frequencies = high_frequencies*tmean*0.045/mean_hf;
```

Varying the values of both images, the hybrid image converge into a more visible result of high frequencies without erasing the low frequencies.

The selection of the images to be added as high_frequency, is not aleatory. The image selection to high - low frequency is based on the color predominance, if the images has different color, the mix is trickier than the images with a same color domain.

3.1.1 Cat - Dog



3.1.2 Bill Gates - Carmen de Mairena

3.1.3 Einstein - Marilyn