

Project 1 Writeup

No one

November 4, 2017

Data Processing

In 1-D convolution, we need to extend two arrays operated on. We can extend with zeros or something else depending on kinds of convolutions. Hence, we also have to preprocess matrices before we convolve them. We have several choices to do it, 'circular', 'replicate', 'symmetric'. According to the requirements, we use zero padding and symmetric padding to process image matrices and we will talk about their different.

1 Algorithm

1.1 2D Convolution in Spacial Domain

After padding the image matrix, for every pixel of the image, we select a sub-matrix whose center is the current pixel and the size of the matrix is the same as the filter matrix. Multiply them by every corresponding element with that the center of the filter matrix has to be multiplied with the current pixel and take the sum of products. The sum is the color value of the corresponding pixel. If the image is presented as RGB, we could do the same operation in three channels at the same time.

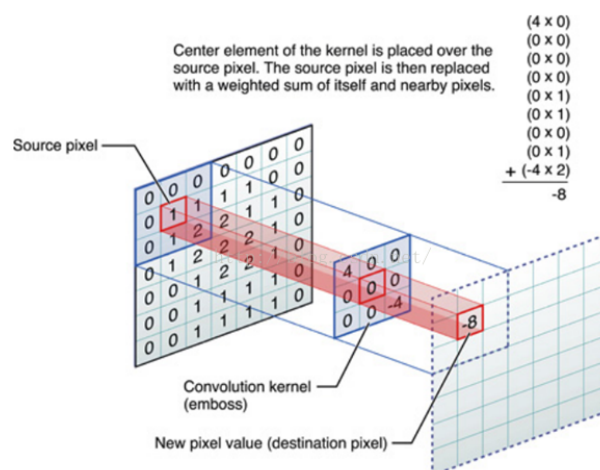


Figure 1: 2D convolution in one channel

1.2 2D Convolution in Frequency Domain

In the Frequency Domain, the convolution operation becomes a multiplication instead, which is faster. In the Frequency Domain, we can use the Fast Fourier Transform to transform the image matrix and filter matrix, then we multiply their transform. Eventually, we use the Inverse Fast Fourier Transform on the product and then we choose the effective part as the output. Moreover, we also need to preprocess the two matrix so that we can easily find the result.

2 Result

Here are the result of aligned images provided and I also find one pair of aligned image and hybrid them.

2.1 Cat and Dog

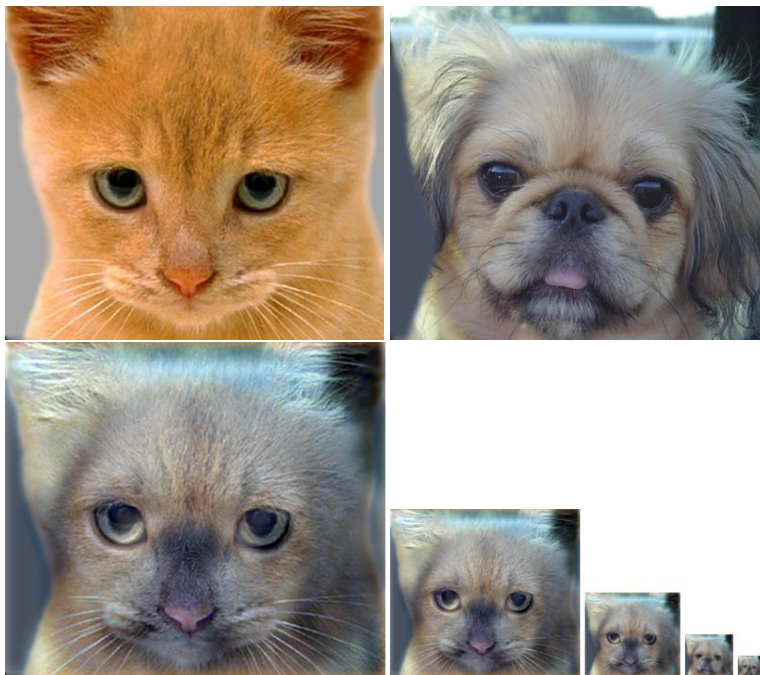


Figure 2: hybrid image 1

2.2 Plane and Bird



Figure 3: hybrid image 2

2.3 Marilyn and Einstein

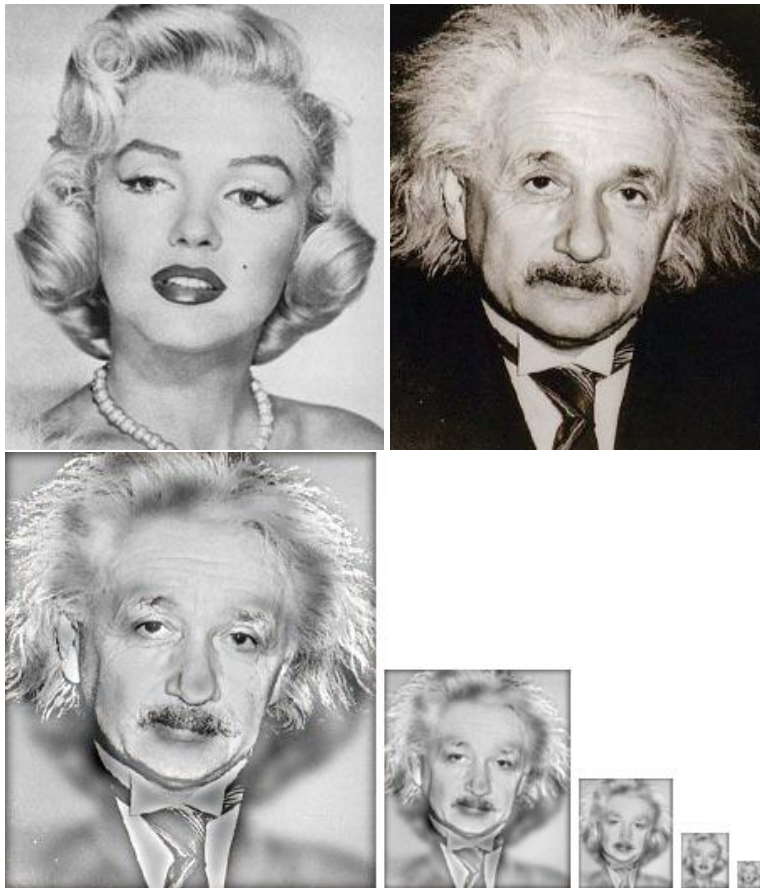


Figure 4: hybrid image 3

2.4 Bicycle and Motorcycle



Figure 5: hybrid image 4

2.5 Fish and Submarine

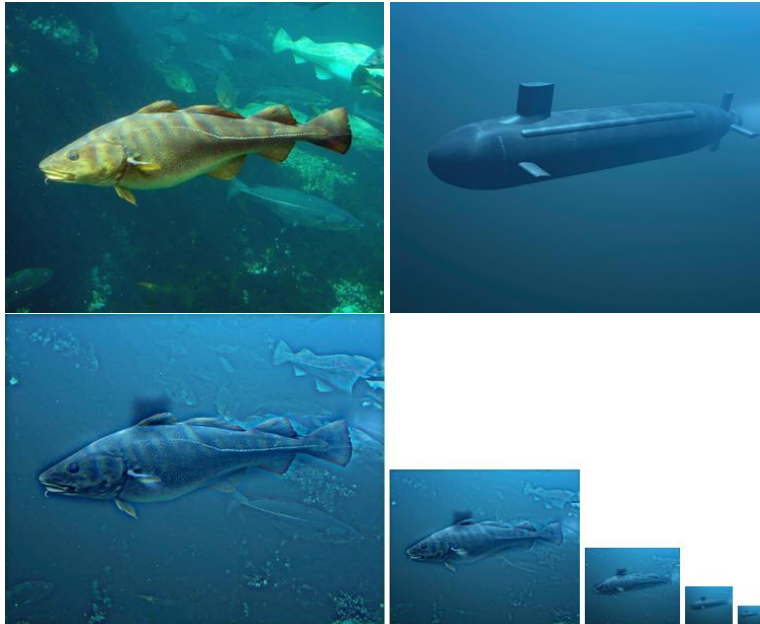


Figure 6: hybrid image 5

2.6 James and Wade



Figure 7: hybrid image 6

3 Speed Test

I test the speed between direct 2D convolution and convolution through FFT. Clearly, the FFT is much faster in different sizes.

Image	Size	Method	Time(seconds)
1	361×410	Direct	3.076361
		FFT	0.053615
2	331×375	Direct	1.528859
		FFT	0.042454
3	265×225	Direct	0.816425
		FFT	0.022070
4	300×453	Direct	1.907693
		FFT	0.038621
5	307×375	Direct	1.569954
		FFT	0.038324
6	450×398	Direct	2.573566
		FFT	0.052624

Table 1: Speed Test in different methods and sizes

4 Bouns

1. Using reflected image content in the code(I called it symmetric padding).
2. Using my own hybrid image in the result.
3. Using FFT to process convolution in the code.

5 Open question

1. The difference between different kinds of padding

My Idea: When we handle the image edges during convolution, we need pad the matrix. Zero padding is obvious the nature way and easy to implement. But if we use zero padding, equivalently we add a black box outside the image. In some way, it looks like not very nature. So we have symmetric padding and cyclic padding. I think the former makes general images more smooth and the later is often used to cyclic image like regular vein.

2. Scaling and Smoothing

My Idea: As we all know, if we shrink a image to some extend, we will lack the precise information about object shapes or else details. Also, smoothing is done the same thing. We ignore details and focus on object contours which is the information provided by low frequencies. So I try to use low pass filter that is gaussian filter to simulate the effect created by shrinking the hybrid image.

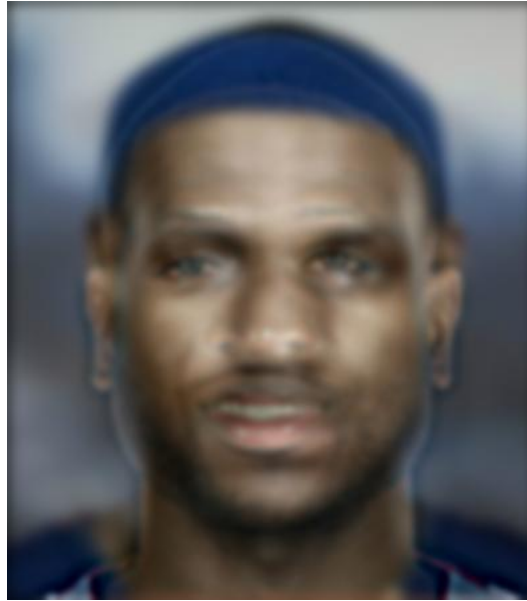


Figure 8: the result of filtering several times