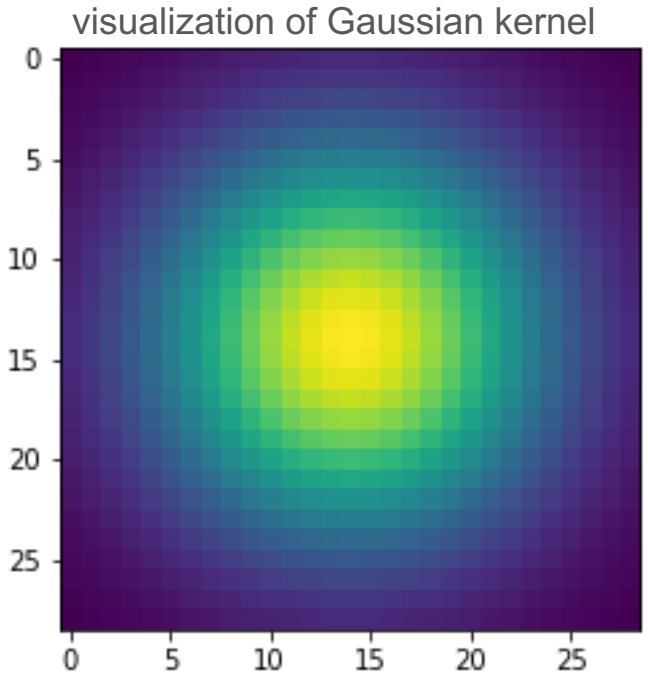
Computer Vision Project 1

[徐简]
[202220013]
[161200063@smail.nju.edu.cn]

Part 1: Image filtering

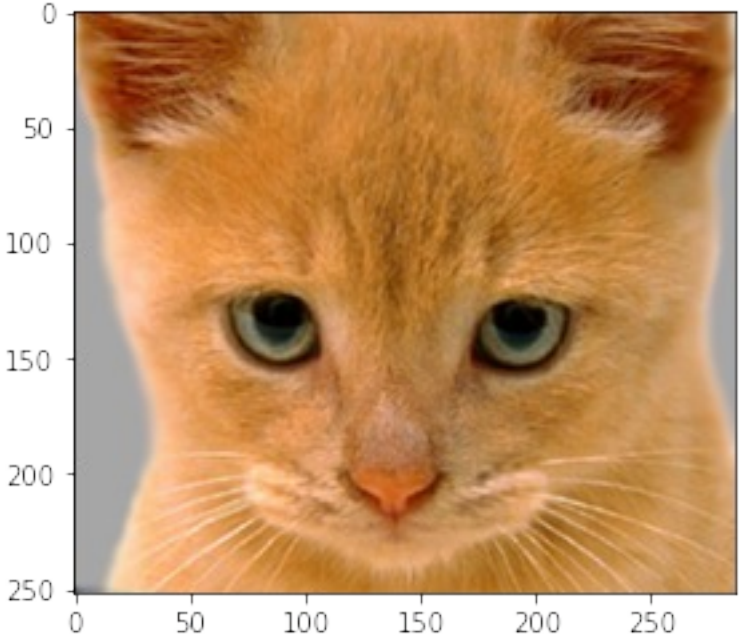


- Add padding to the image and the size of padding should be enough for the first pixel to be in the center of the kernel. So, padding = floor(kernel.size/2)
- Iterate through the image and apply filter to each pixel. pixel grid * filter and calculate the sum, which is the value of the new pixel.

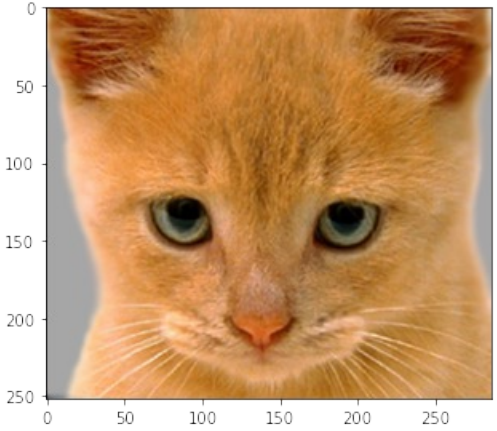
- The key of conv2d is to use the right range of pixel index.

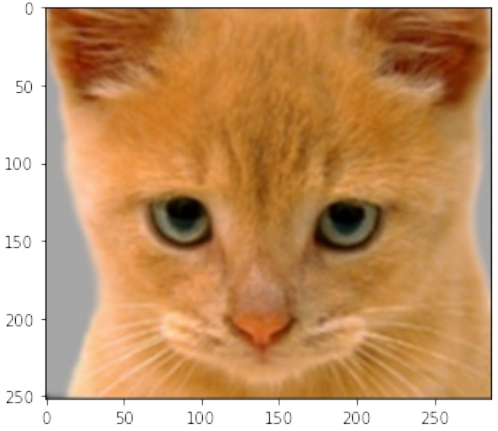
Part 1: Image filtering

Identity filter



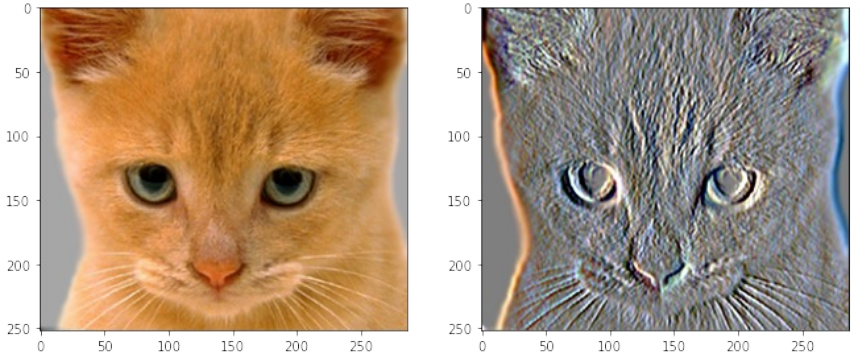
Small blur with a box filter



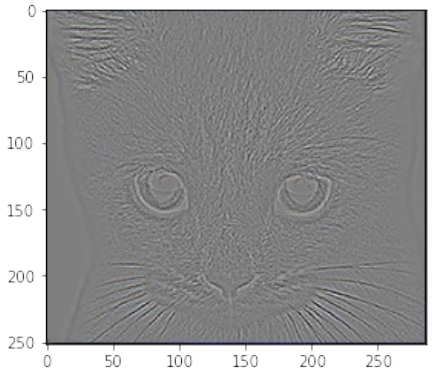


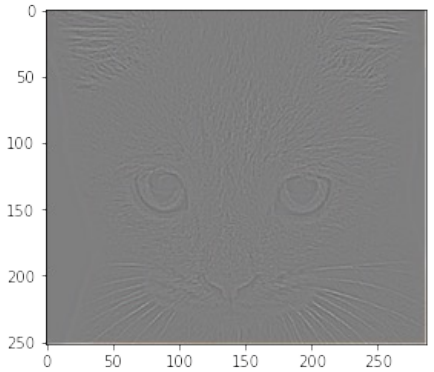
Part 1: Image filtering

Sobel filter



Discrete Laplacian filter





Part 1: Hybrid images

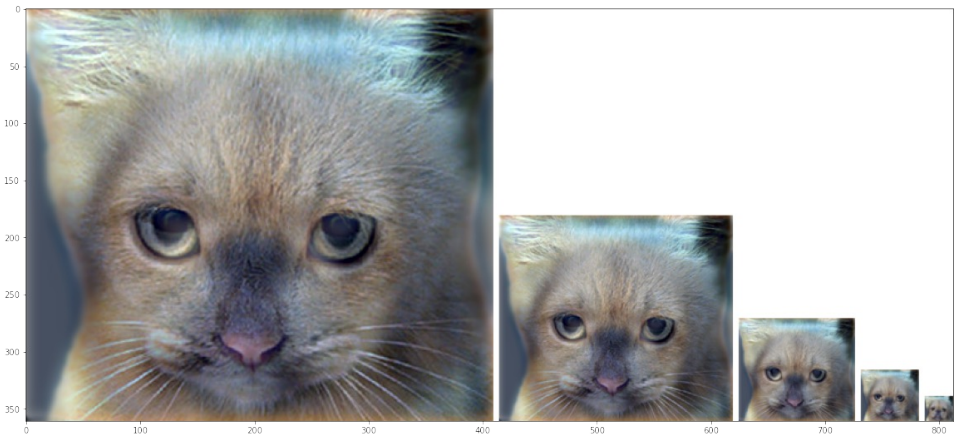
(1) Apply the filter on both images to get the low frequency parts

(2) The high frequency part of the image can be got by removing the low frequency part from the original one. High = Original – Low

(3) Then, add the low part and the high part together.

(4) Make sure to clip the hybrid image, so that the value of index is in range(0,1)

Cat + Dog



Cutoff frequency: 7

Part 1: Hybrid images

Motorcycle + Bicycle

Cutoff frequency: 5

Plane + Bird

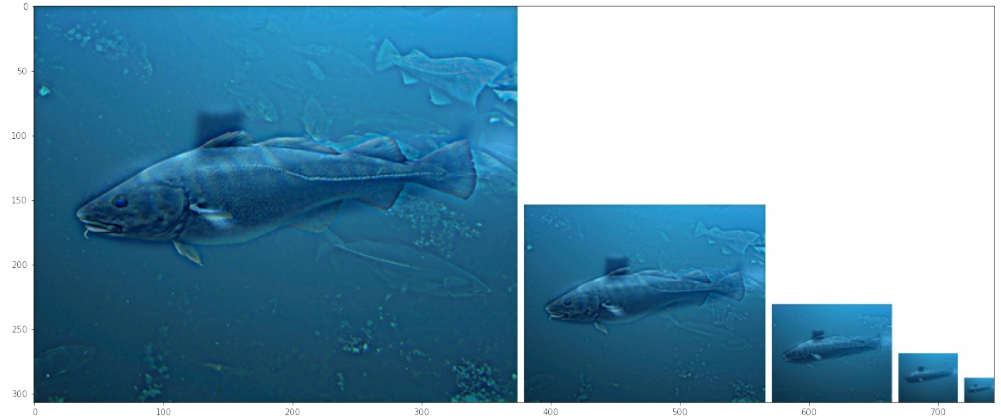
Cutoff frequency:6

Part 1: Hybrid images

Einstein + Marilyn

Cutoff frequency: 4

Submarine + Fish



Cutoff frequency:3

Part 2: Hybrid images with PyTorch

Cat + Dog

Motorcycle + Bicycle

Part 2: Hybrid images with PyTorch

Plane + Bird

Einstein + Marilyn

Part 2: Hybrid images with PyTorch

Submarine + Fish

Part 1 vs. Part 2

Part 1: 9.172 seconds

Part 2: 0.131 seconds

Part2 is much faster.

Part 3

[Consider a 1-channel 5x5 image and a 3x3 filter. What are the output dimensions of a convolution with the following parameters?
Stride = 1, padding = 0→(3,3,1)
Stride = 2, padding = 0→(2,2,1)
Stride = 1, padding = 1→(5,5,1)
Stride = 2, padding = 1→ (3,3,1)

[What are the input & output dimensions of the convolutions of the dog image and a 3x3 filter with the following parameters:
Stride = 1, padding = 0
Stride = 2, padding = 0
Stride = 1, padding = 1
Stride = 2, padding = 1

dog(1,3,361, 410)
(1)(359,408,3)
(2)(180,204,3)
(3)(361,410,3)
(4)(181,205,3)

Part 3

[How many filters did we apply to the dog image?]

12 filters in total
4*3

[Why do the output dimensions adhere to the equations given in the instructions handout?]

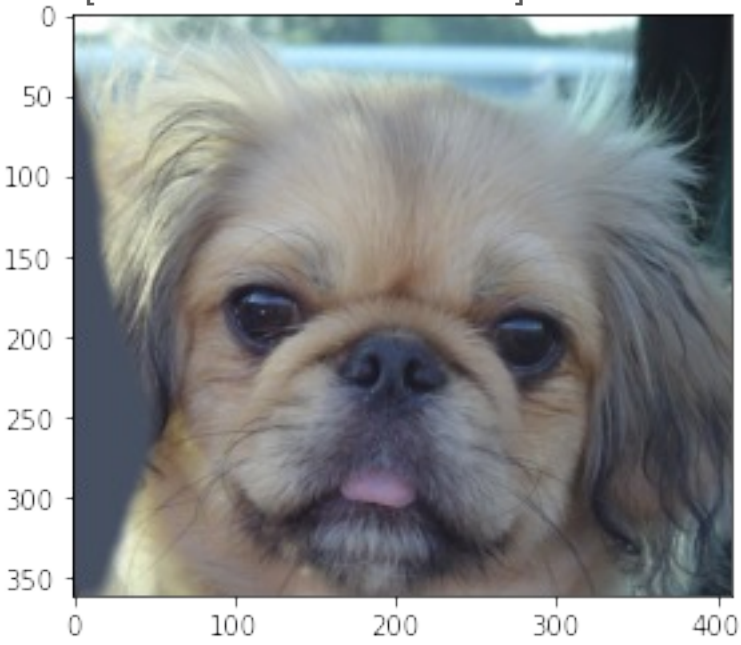
Kernel(N,d1/g,k,k)
So the output dimension equals to N.
In part 3, N is the number of filters stacked together.

Part 3

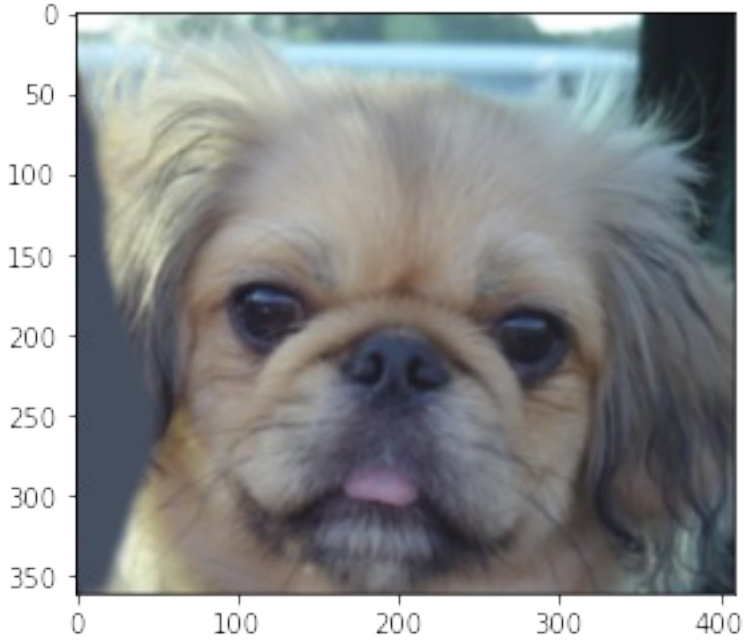
[What is the intuition behind this equation?]
The output size depends on the stride and padding. The larger the stride, the smaller the output. The larger the padding, the larger the output.
Besides, we can apply different filters on one image at the same time. We simply need to stack filters together and the channels of the output will change, which indicates the result of different filters.

Part 3

[insert visualization 0 here]

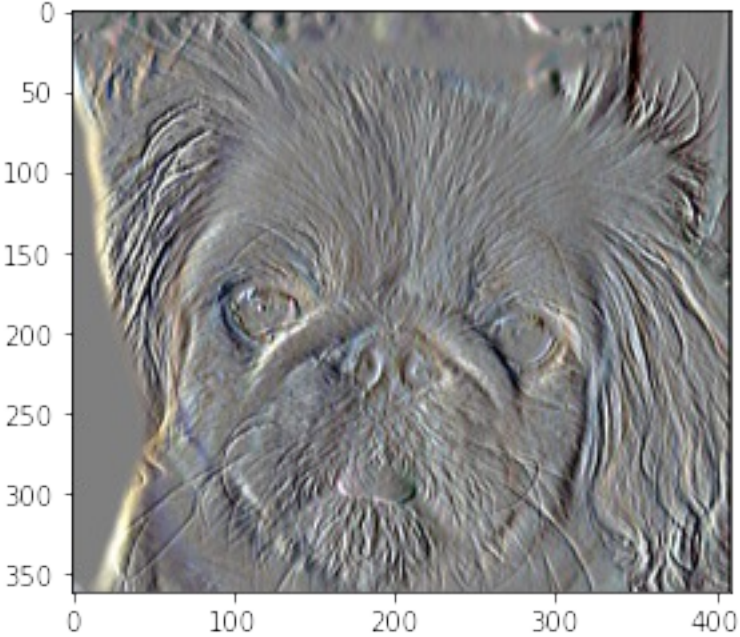


[insert visualization 1 here]

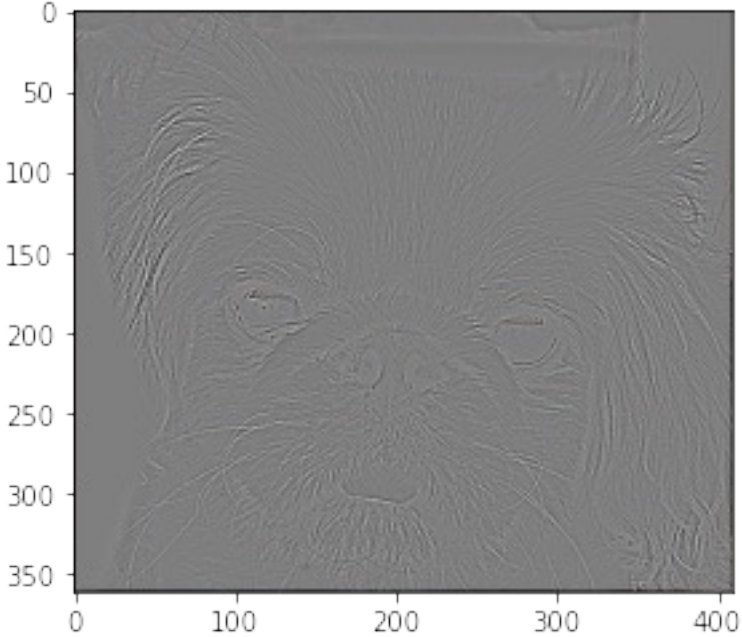


Part 3

[insert visualization 2 here]



[insert visualization 3 here]



Conclusion

[How does varying the cutoff frequency value or swapping images within a pair influences the resulting hybrid image?]

The cutoff frequency determines the ration of low frequency part and high frequency part. The larger the cutoff frequency, the more the hybrid image looks like the high frequency part. So, in order to have a nice hybrid image, you have to try different cutoff frequencies.

Besides, which image to reserve the low frequency part and which to reserve the high frequency part is also important. Images which can be classified by basic outline should contribute their low frequency parts. On the contrary, images can only be classified by details should keep their high frequency parts. Therefore, swapping images may also bring out a nicer hybrid image.