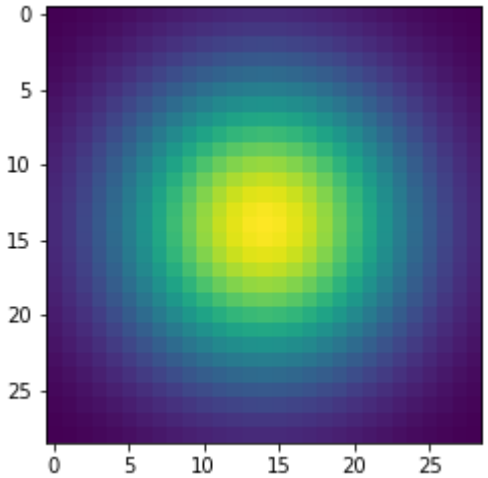
Computer Vision Project 1

[刘畅]
[221900071]
[221900071@smial.nju.edu.cn]

Part 1: Image filtering

[insert visualization of Gaussian kernel from proj1.ipynb here]



[Describe your implementation of my_conv2d_numpy() in words. Make sure to discuss padding, and the operations used between the filter and image.]

输入:image(m,n,c) filter(p,q)

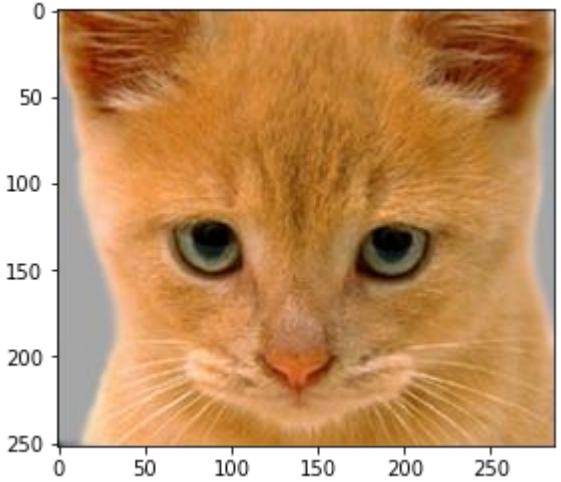
填充：在图像的边缘区域采用0填充（version2中使用镜像填充），且选取特定的填充宽度和高度以确保填充后图像的宽高不发生变化。

卷积：逐像素点逐通道遍历，按照卷积的计算规则对原图像子矩阵和卷积核心做卷积运算。

Part 1: Image filtering

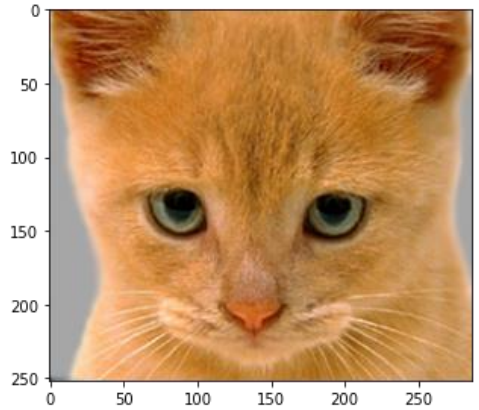
Identity filter

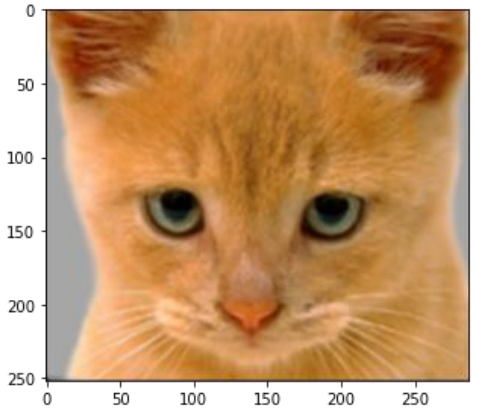
[insert the results from proj1.ipynb using 1b_cat.bmp with the identity filter here]



Small blur with a box filter

[insert the results from proj1.ipynb using 1b_cat.bmp with the box filter here]

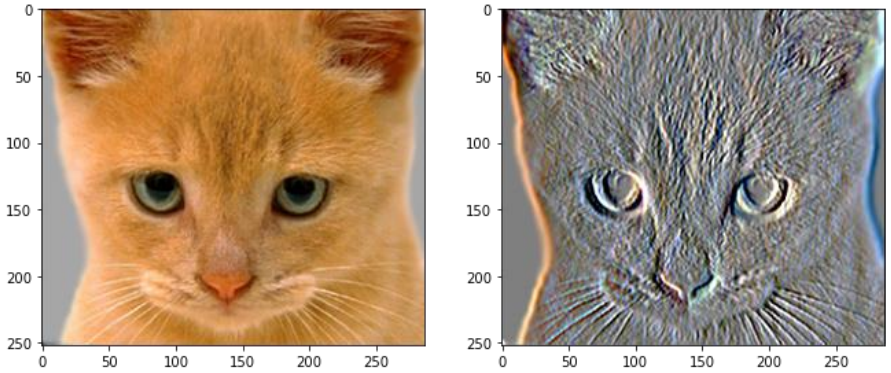




Part 1: Image filtering

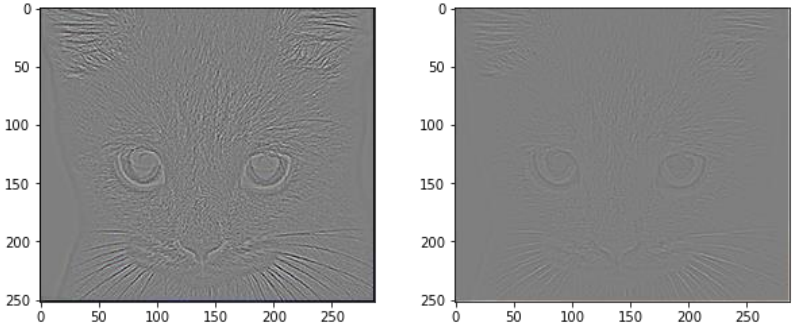
Sobel filter

[insert the results from proj1.ipynb using 1b_cat.bmp with the Sobel filter here]



Discrete Laplacian filter

[insert the results from proj1.ipynb using 1b_cat.bmp with the discrete Laplacian filter here]



Part 1: Hybrid images

[Describe the three main steps of create_hybrid_image() here. Explain how to ensure the output values are within the appropriate range for matplotlib visualizations.]

1.低频和高频分离

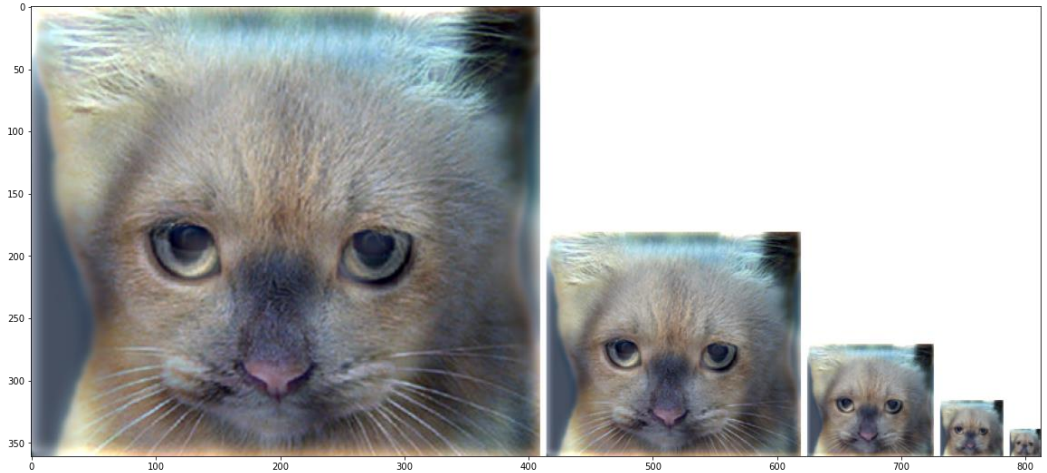
2.将低频和高频图像进行混合

3.将输出像素值映射到[0,1]区间内以便可视化

使用了np.clip方法将像素值映射到了[0,1]区间内，在此基础上使用提供的vis_image_scales_numpy方法进行可视化

Cat + Dog

[insert your hybrid image here]



Cutoff frequency: [7]

Part 1: Hybrid images

Motorcycle + Bicycle

[insert your hybrid image here]

Cutoff frequency: [7]

Plane + Bird

[insert your hybrid image here]

Cutoff frequency: [7]

Part 1: Hybrid images

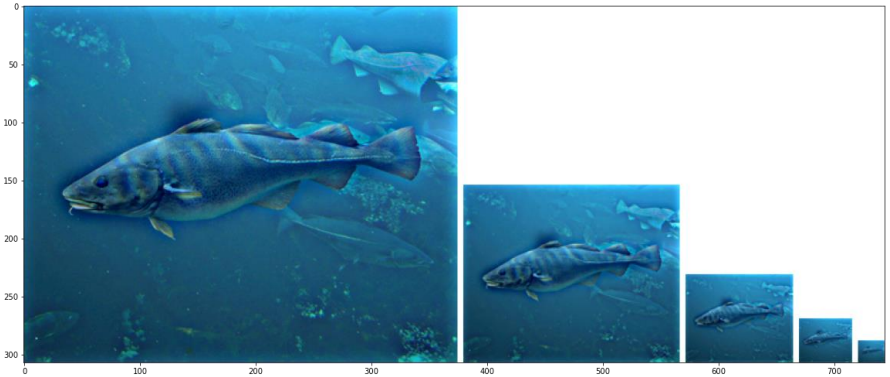
Einstein + Marilyn

[insert your hybrid image here]

Cutoff frequency: [7]

Submarine + Fish

[insert your hybrid image here]



Cutoff frequency: [7]

Part 2: Hybrid images with PyTorch

Cat + Dog

[insert your hybrid image here]

Motorcycle + Bicycle

[insert your hybrid image here]

Part 2: Hybrid images with PyTorch

Plane + Bird

[insert your hybrid image here]

Einstein + Marilyn

[insert your hybrid image here]

Part 2: Hybrid images with PyTorch

Submarine + Fish

[insert your hybrid image here]

Part 1 vs. Part 2

[Compare the run-times of Parts 1 and 2 here, as calculated in proj1.ipynb. Which method is faster?]

使用Pytorch的方法速度更快

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
Stride = 2, padding = 0? => 2x2
Stride = 1, padding = 1? => 5x5
Stride = 2, padding = 1?] => 3x3

[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 Image: 361 × 410 × 3
Stride = 1, padding = 0 => 359 × 408 × 3
Stride = 2, padding = 0 => 180 × 204 × 3
Stride = 1, padding = 1 => 361 × 410 × 3
Stride = 2, padding = 1 => 181 × 205 × 3

Part 3

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

12种

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

1. **I-K**表示原图中能被完整覆盖的剩余空间
2. **2*Padding***：左右/上下各加一圈增加的有效空间
3. **除以Stride :**得到跳步的步数
4. **+1**：加上初始位置

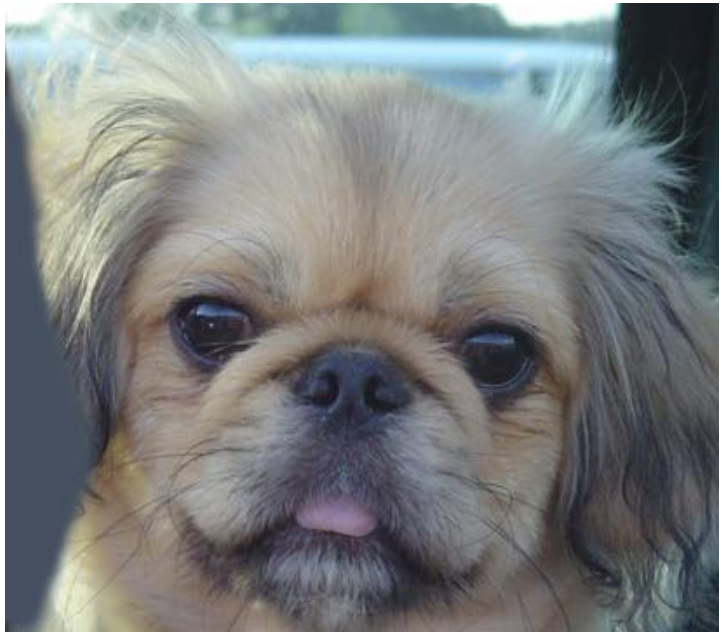
Part 3

[What is the intuition behind this equation?]

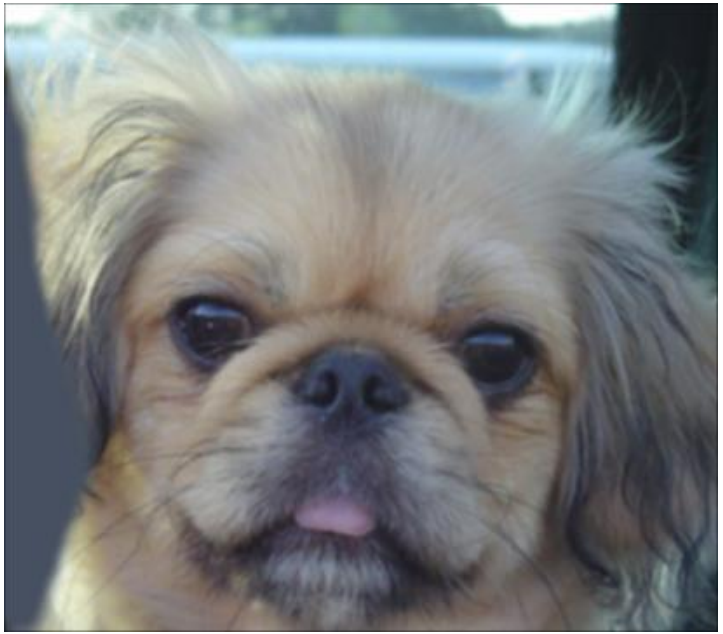
使用stride=1,padding= floor((K-1)/2)时,能够使得卷积后图形尺寸较卷积前不变

Part 3

[insert visualization 0 here]

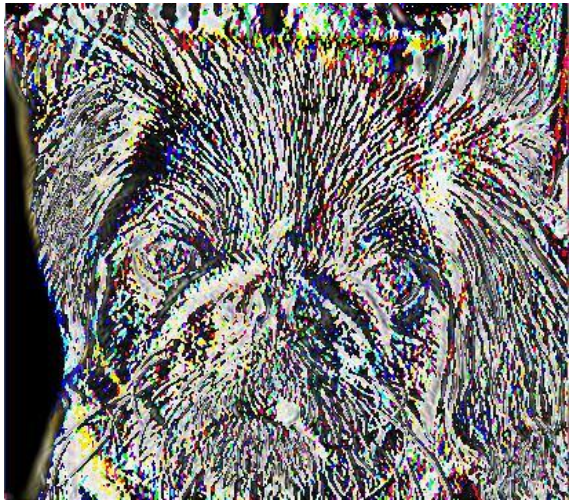


[insert visualization 1 here]

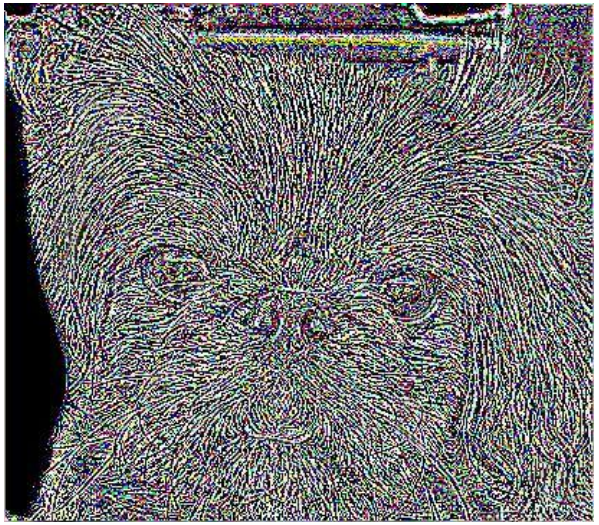


Part 3

[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?]

增大截止频率：图像保留更多高频信息，混合图像会更清晰

减小截止频率：图像会更模糊，但是更平滑

交换图像的顺序会导致图像在近处和远处的相对视觉位置交换