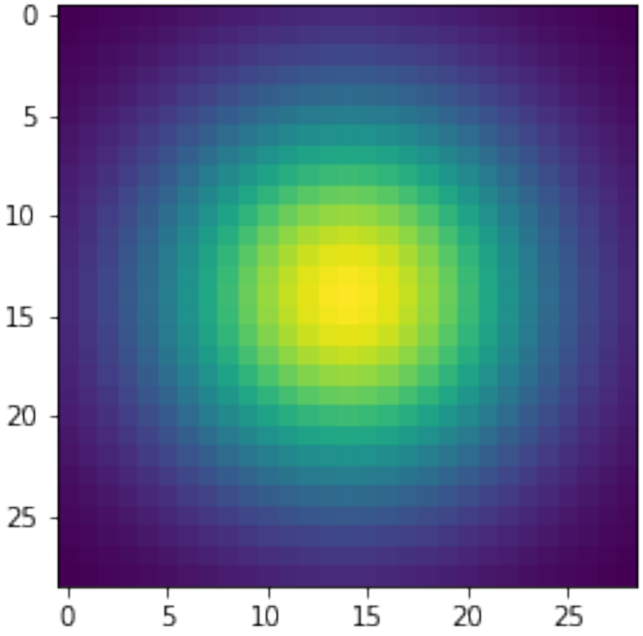
CS 4476 Project 1

Jie Lyu
jie.lyu@gatech.edu
jlyu31
903329676

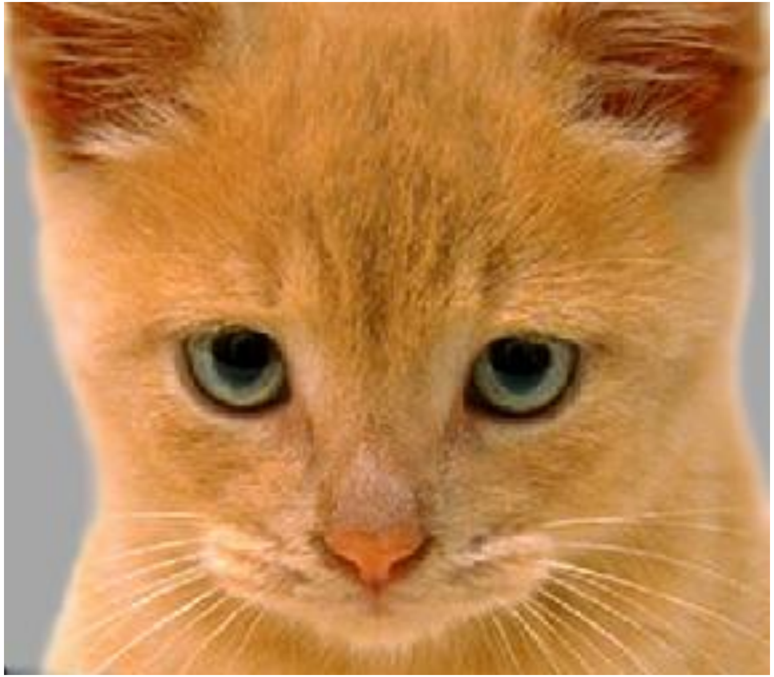
Part 1: Image filtering



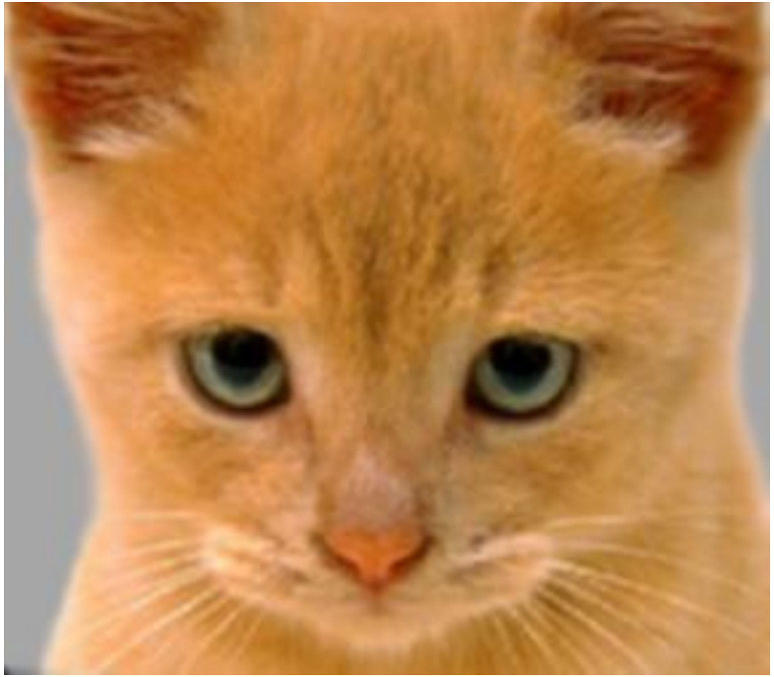
2d Gaussian kernel

I first generate a 1d Gaussian array using numpy.fromfunction() and a helper method I wrote that does the Gaussian calculation using the standard 1D Gaussian formula. Then I used numpy.outer to find the outer product of the 1D array with itself, which is a 2d Gaussian array

Part 1: Image filtering

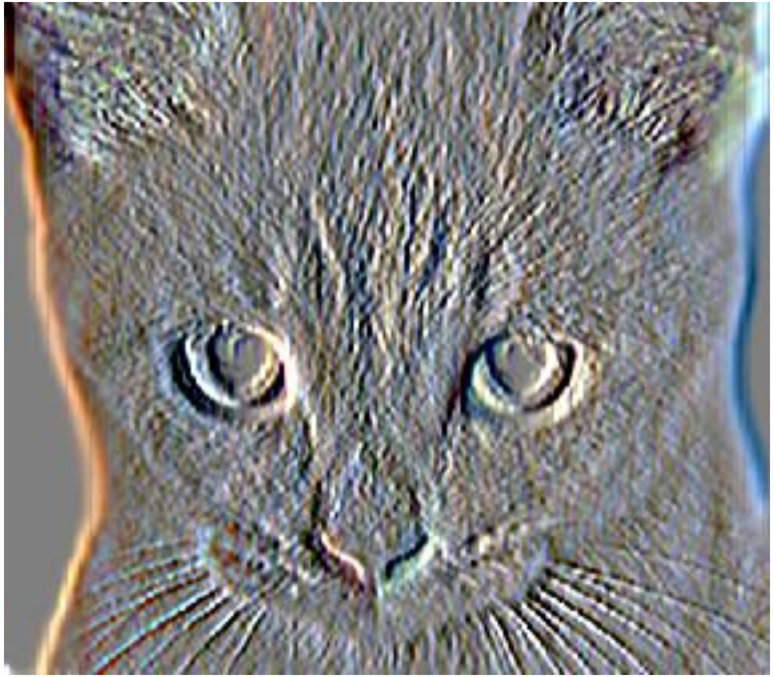


Identity filter

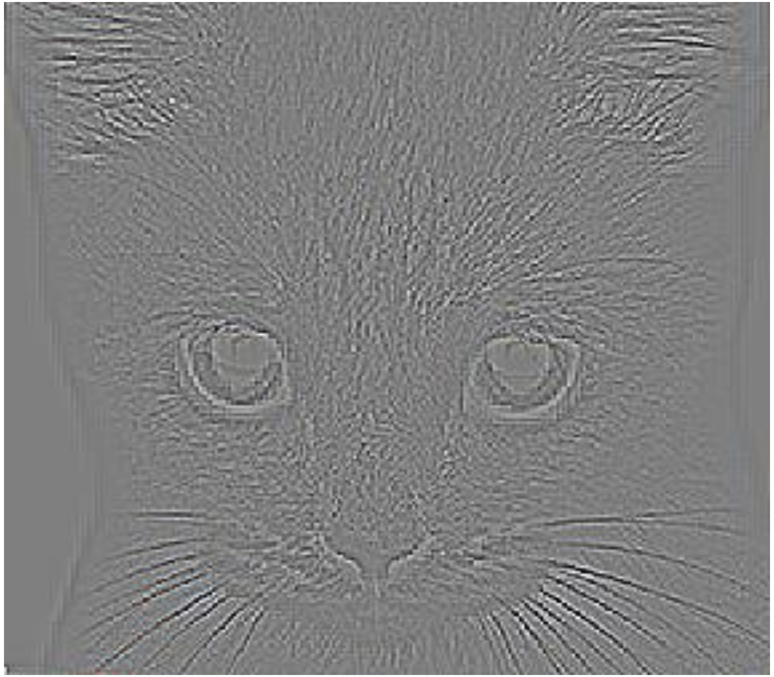


Small blur with a box filter

Part 1: Image filtering



Sobel filter

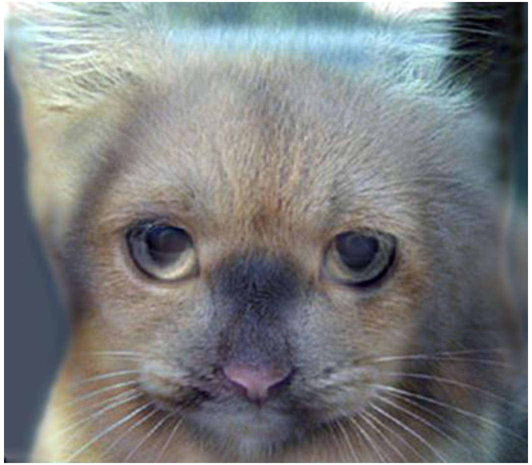


Discrete Laplacian filter

Part 1: Hybrid images

I used my_imfilter() to get the low frequency filtered version of both images a and b, and subtracted the filtered image b one from the original image b to get the high frequency one. Then I added the low frequency version of image a and high frequency version of image b to get the final hybrid image. I used numpy.clip() to make sure the pixel values are between 0 and 1

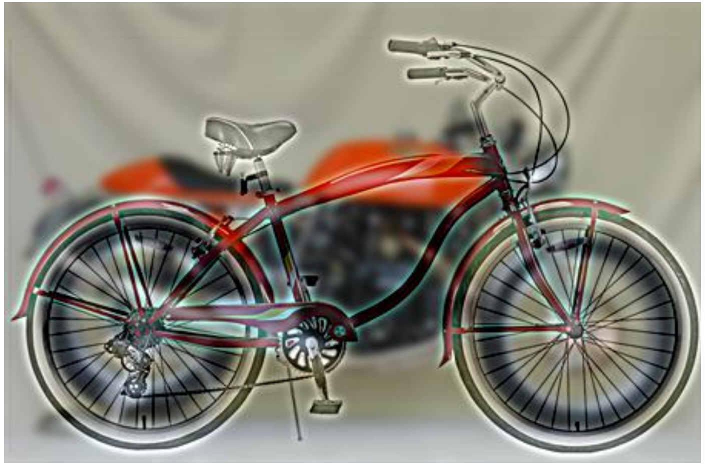
Cat + Dog



Cutoff frequency: 5

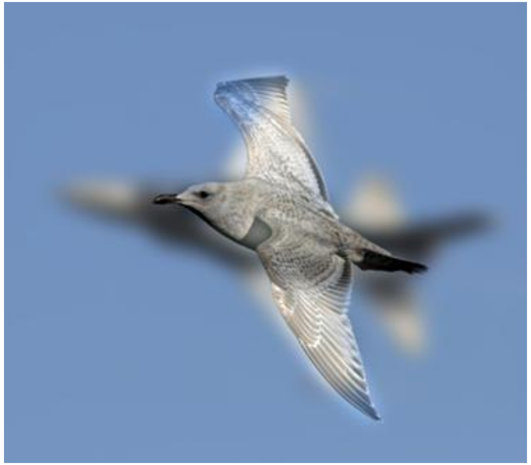
Part 1: Hybrid images

Motorcycle + Bicycle



Cutoff frequency: 3

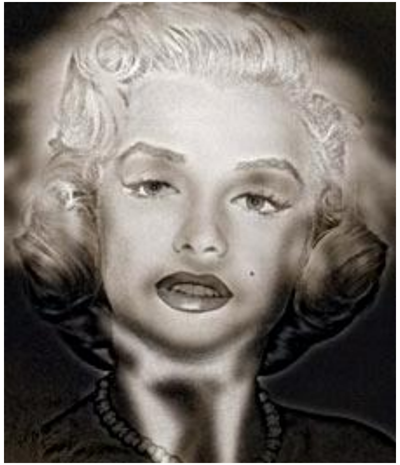
Plane + Bird



Cutoff frequency: 7

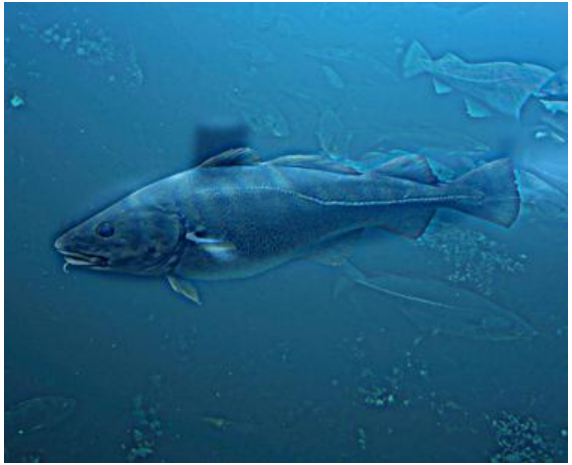
Part 1: Hybrid images

Einstein + Marilyn



Cutoff frequency: 3

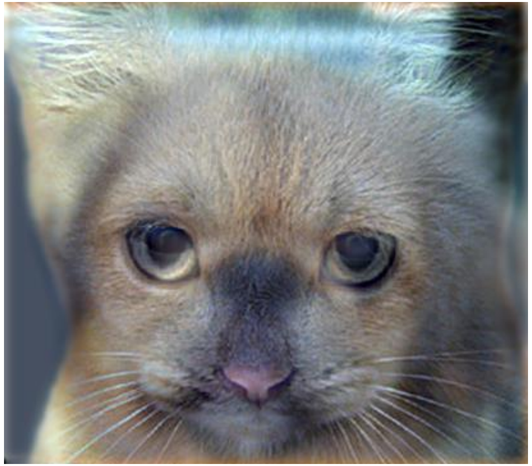
Submarine + Fish



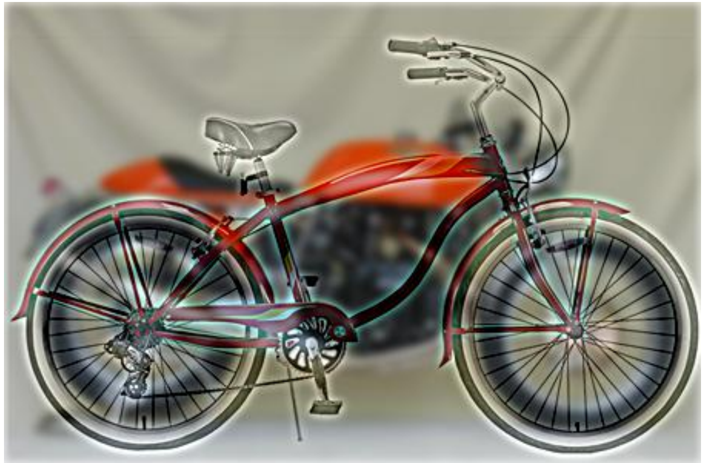
Cutoff frequency: 3

Part 2: Hybrid images with PyTorch

Cat + Dog

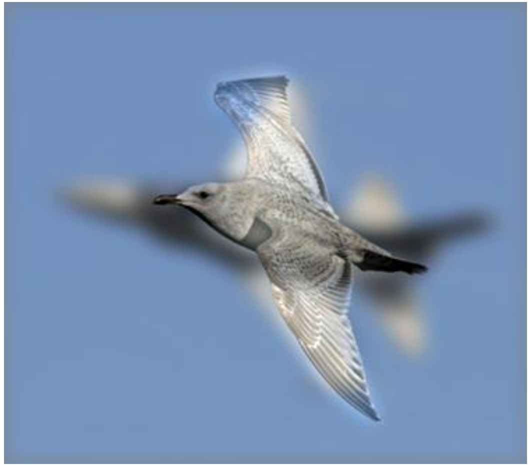


Motorcycle + Bicycle

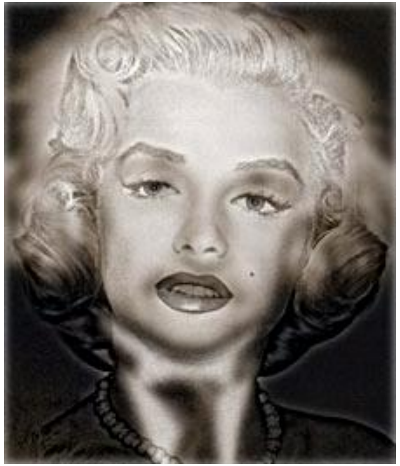


Part 2: Hybrid images with PyTorch

Plane + Bird

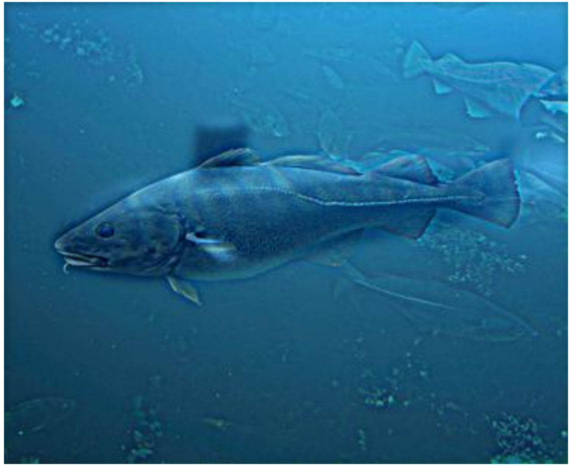


Einstein + Marilyn



Part 2: Hybrid images with PyTorch

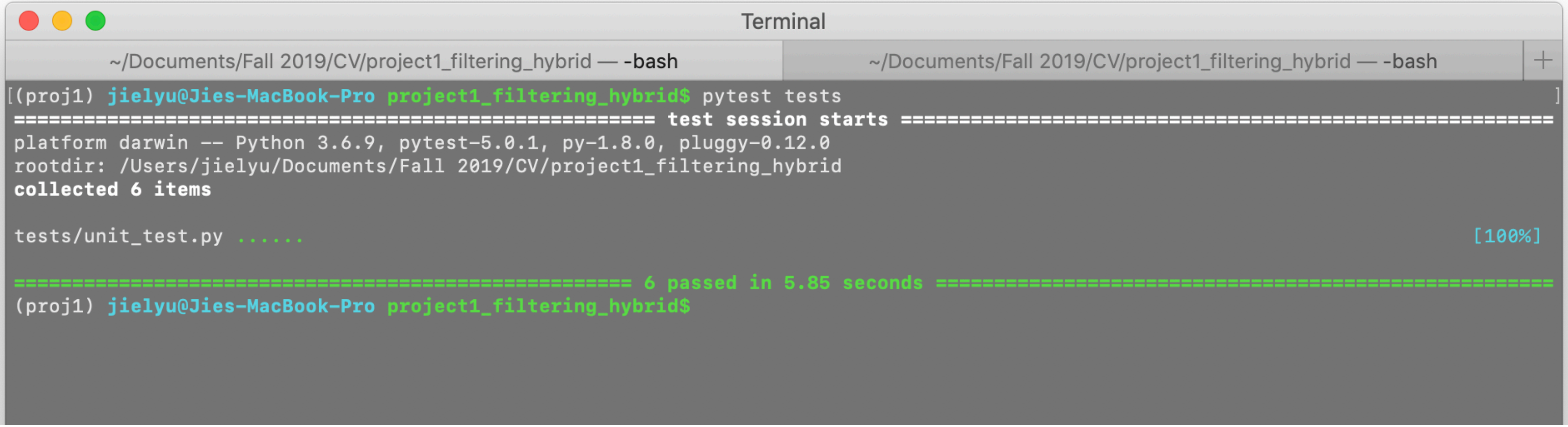
Submarine + Fish



Part 1 vs. Part 2

Part 1 took 9.276 seconds and part 2 took 1.677 second. Part 2 is 82% faster. To me that means better.

Tests



Terminal
~/Documents/Fall 2019/CV/project1_filtering_hybrid — -bash
~/Documents/Fall 2019/CV/project1_filtering_hybrid — -bash

```
(proj1) jielyu@Jies-MacBook-Pro project1_filtering_hybrid$ pytest tests
======================================================= test session starts ========================================================
platform darwin -- Python 3.6.9, pytest-5.0.1, py-1.8.0, pluggy-0.12.0
rootdir: /Users/jielyu/Documents/Fall 2019/CV/project1_filtering_hybrid
collected 6 items

tests/unit_test.py ......                                                                                                    [100%]

===================================================== 6 passed in 5.85 seconds =====================================================
(proj1) jielyu@Jies-MacBook-Pro project1_filtering_hybrid$
```

Conclusions

With a default cutoff frequency at 7, some high frequency images shows too much edge (e.g. Einstein + Marilyn), thus dominating the hybrid images and making people hard to ignore it. Lowering the cutoff frequency in those cases will produce a better result. Also I found it better to set the frequency to 5 for the cat and dog case. Rods and cones works different for everybody, so for me this frequency works the best. The biggest challenge has been learning to work with numpy, but it is also rewarding since it is an awesome tool.

Thank you for reading!