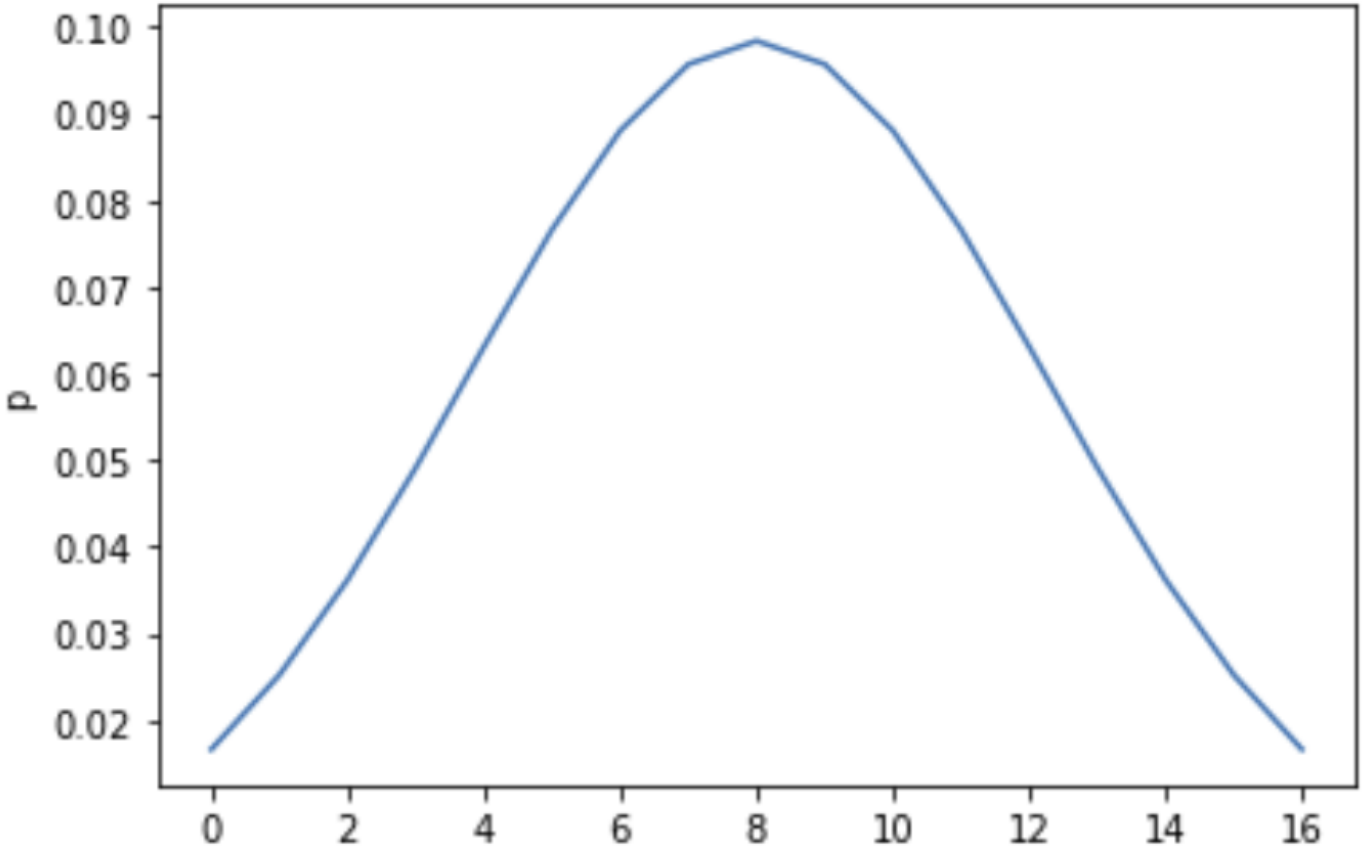
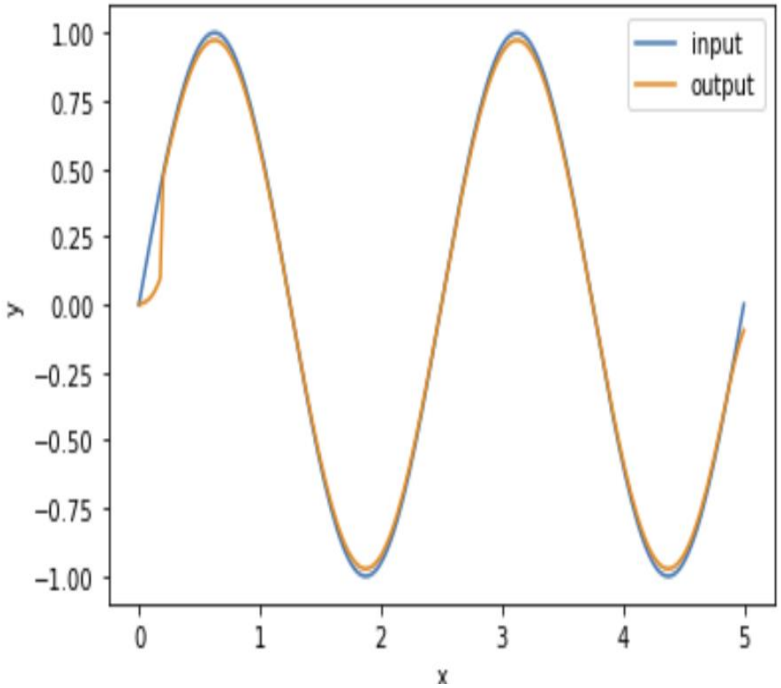
CS 6476 Project 1

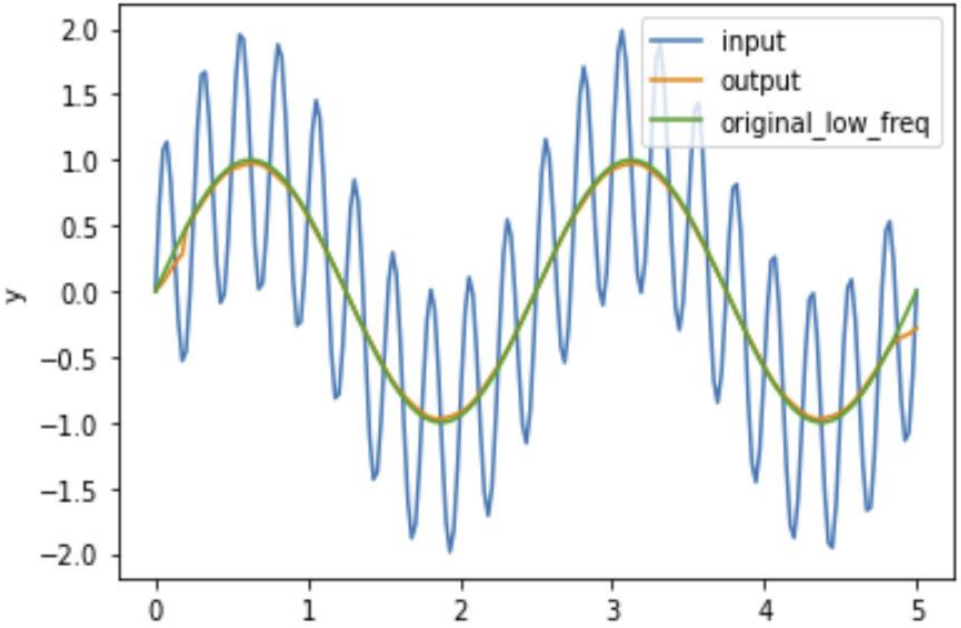
Safin Salih
Ssalih6@gatech.edu
ssalih6
902111076

Part 1: 1D Filter

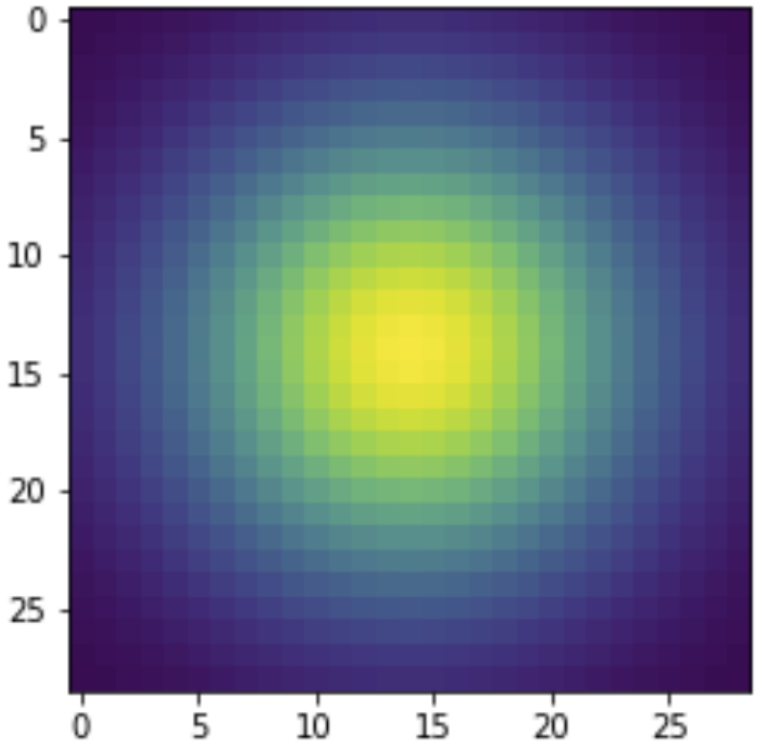


Part 1: 1D Filter





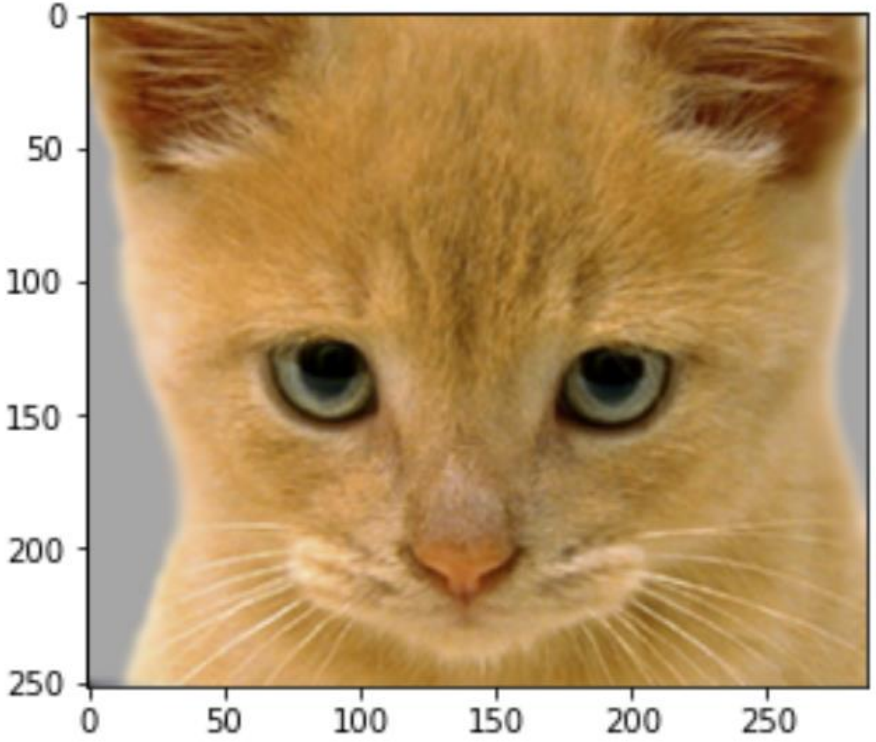
Part 2: Image Filtering



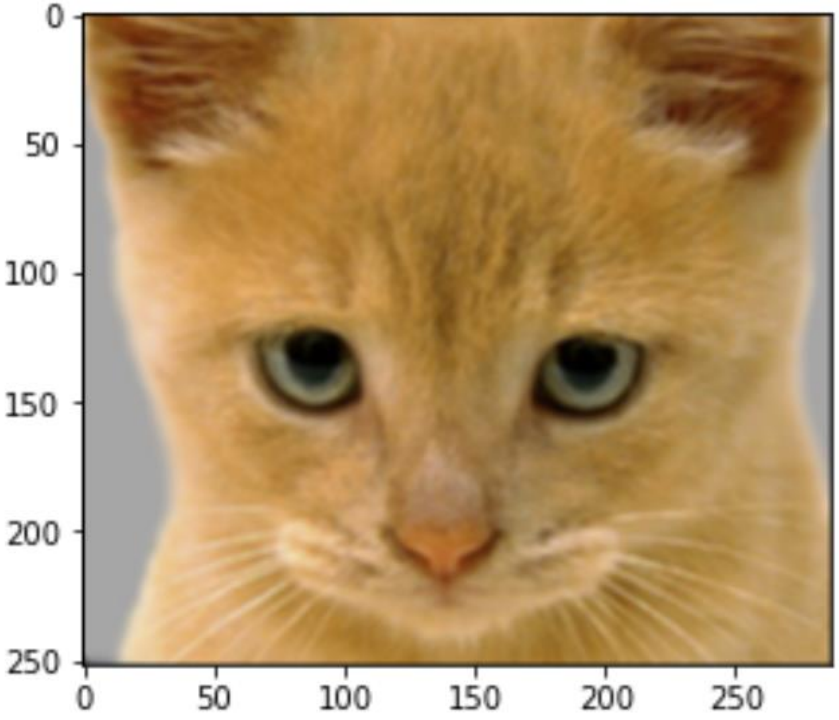
I created a function similar to 1D gaussian, took the outout vector, which would be a 1D vector, and since it's symmetric I took the outer product of that same vector which would create a N by N kernel Matrix. Then I divided each element by the sum of matrix, Hence, when you sum the kernel, it equals to one.

Part 2: Image filtering

Identity filter

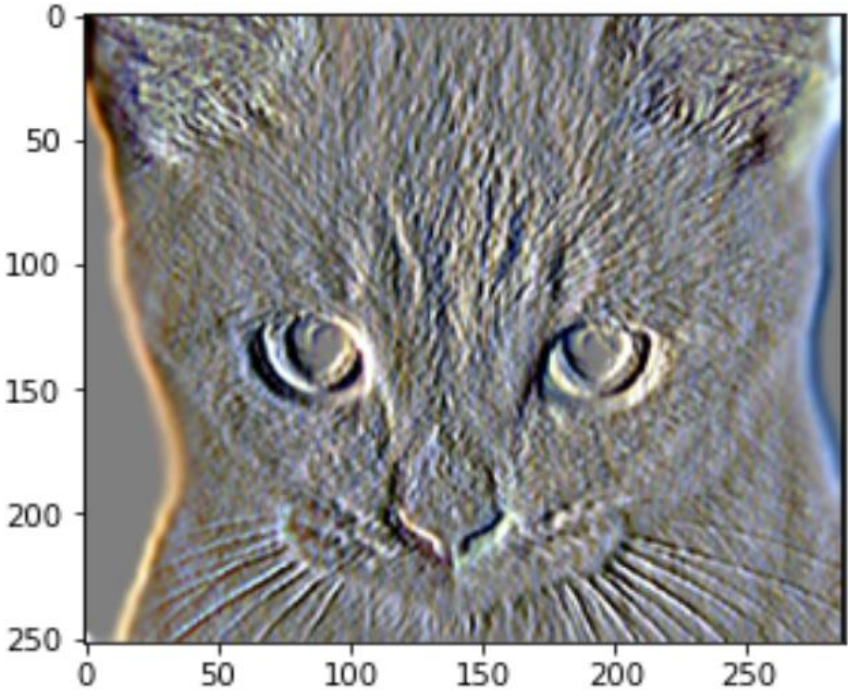


Small blur with a box filter

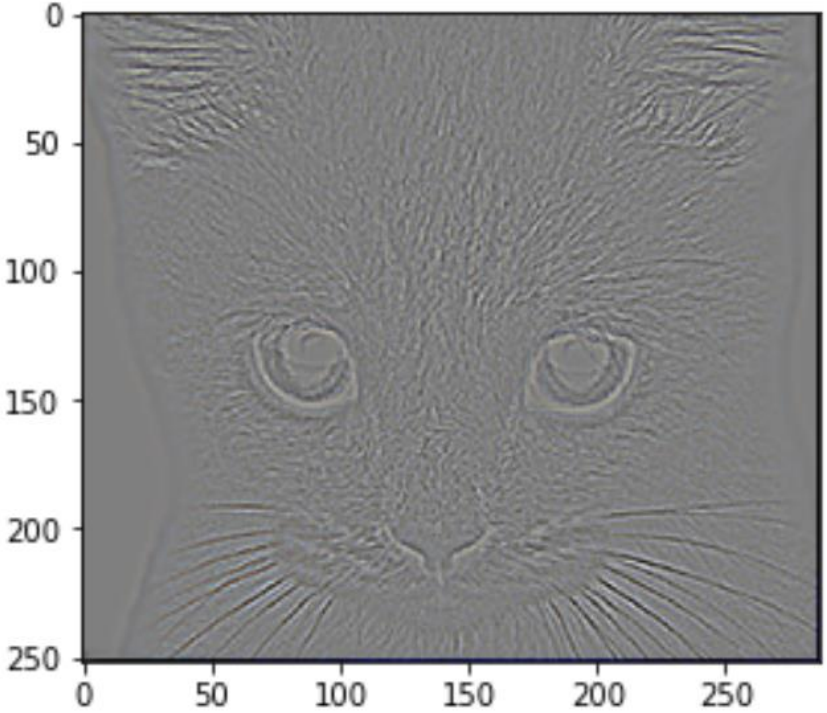


Part 2: Image filtering

Sobel filter



Discrete Laplacian filter



Part 2: Hybrid images manually using Pytorch

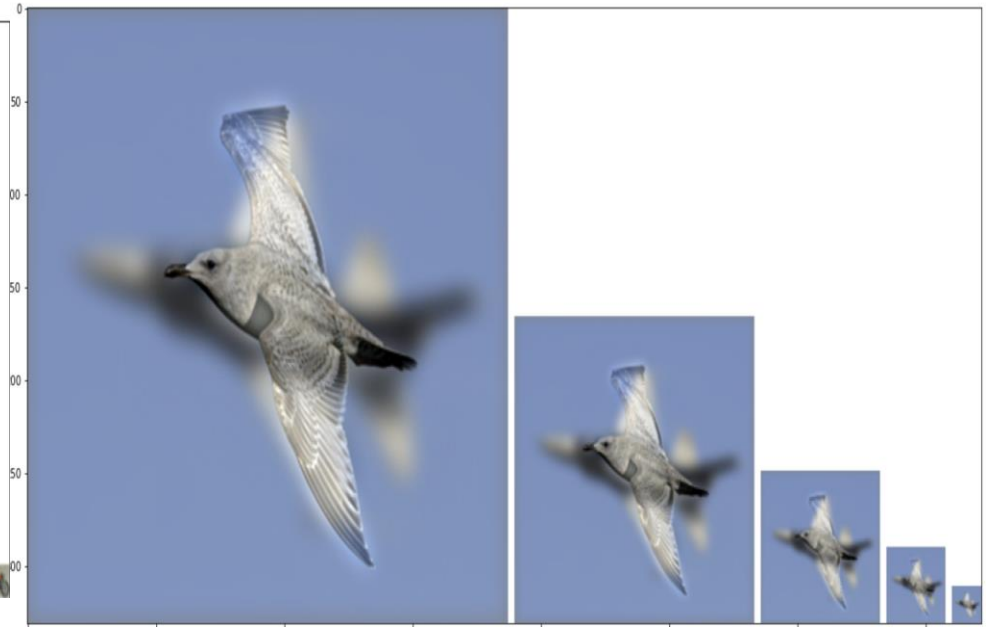
First to get the high frequency image, first get the low frequency of both images by using my_imfilter(). Then by subtracting second image by low frequency of that image, you get the high frequency . Afterwards, I used torch.clamp to make pixels be between 0 and 1.

Part 2: Hybrid images manually using Pytorch

Motorcycle + Bicycle

7

Plane + Bird



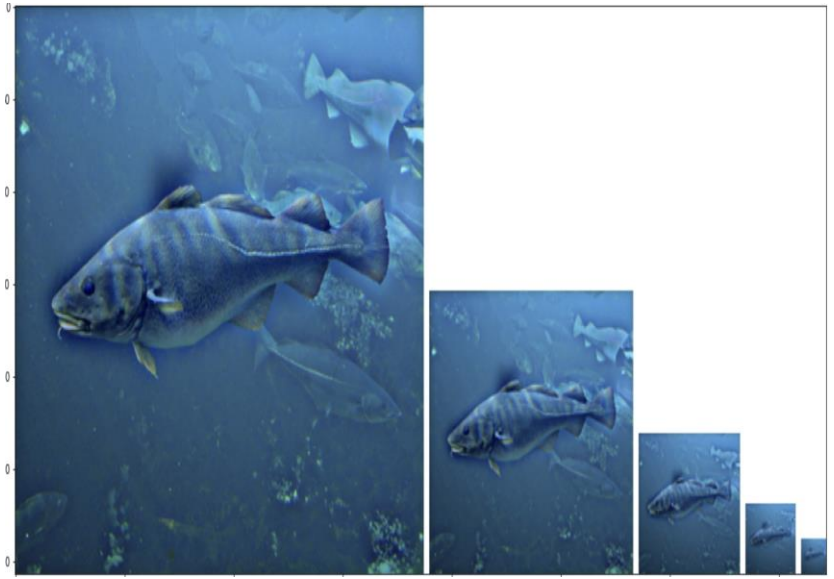
7

Part 2: Hybrid images manually using Pytorch

Einstein + Marilyn

7

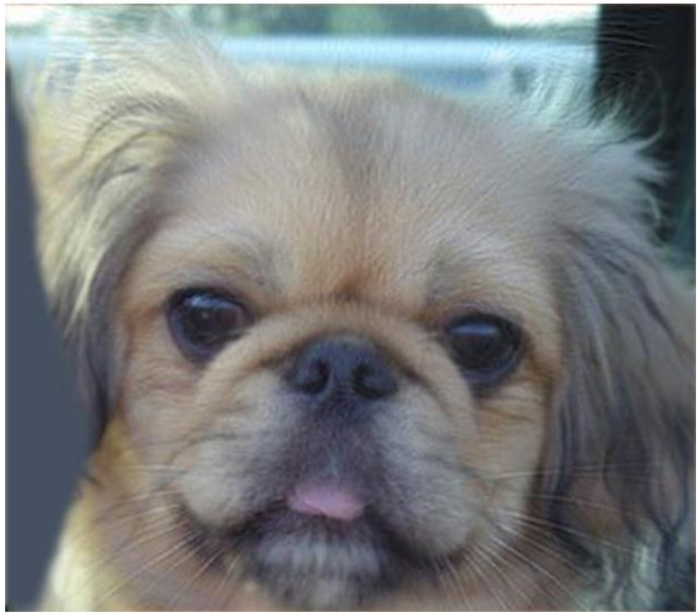
Submarine + Fish



7

Part 3: Hybrid images with PyTorch operators

Cat + Dog



1

Motorcycle + Bicycle

5

Part 3: Hybrid images with PyTorch operators

Plane + Bird

10

Einstein + Marilyn

15

Part 3: Hybrid images with PyTorch operators

Submarine + Fish

Part 2 vs. Part 3

The run-time took for me around 24.231 seconds, while Part 2 took roughly 0.718 seconds. So method two was much faster.

Tests

```
FAILED proj1_unit_tests\test_dft.py::test_dft - NotImplementedError: `my_dft` function in `dft.py` needs to be implem...
==================================== 3 failed, 10 passed, 3 warnings in 13.12s ====================================
(proj1) C:\Users\safin\Downloads\New folder\proj1_release\proj1_code>
```

Pairs: This question must be answered individually

Conclusions

I learned about gaussian blur, and concepts surrounding kernel filter . The parameters I played around with were the cutoff std, and around 5 or 6, typically lower frequencies give a more feasible understanding. When trying to merge images, the sharpness of an images truly makes a difference, so you have to becareful how you set your std values. The biggets challenge was learning some of the functionalities of torch/numpy and just didn’t have enough time to get to the extra credit.

Extra Credit

Image Filtering using DFT

<insert visualization of the DFT filtered 6a_dog.bmp and 6b_cat.bmp from proj1.ipynb here>

Describe your implementation in words.

Add some cool hybrid images!