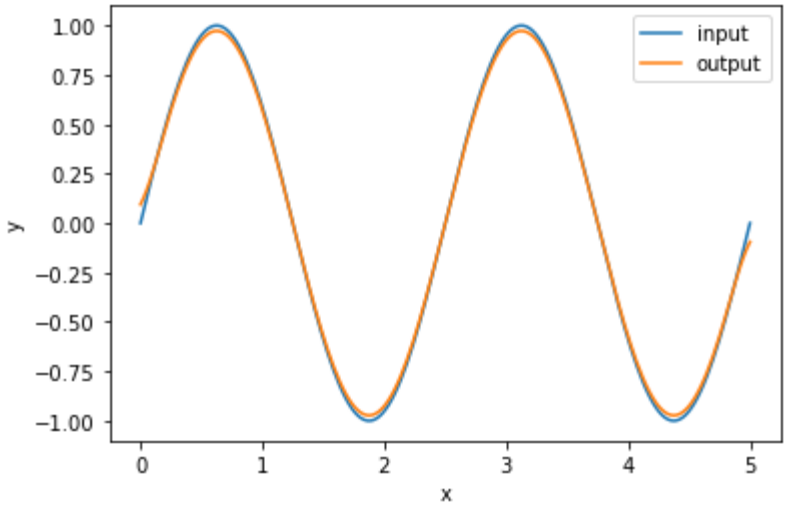
CS 6476 Project 1

Karan Sarkar
ksarkar9@gatech.edu
ksarkar9
903569592

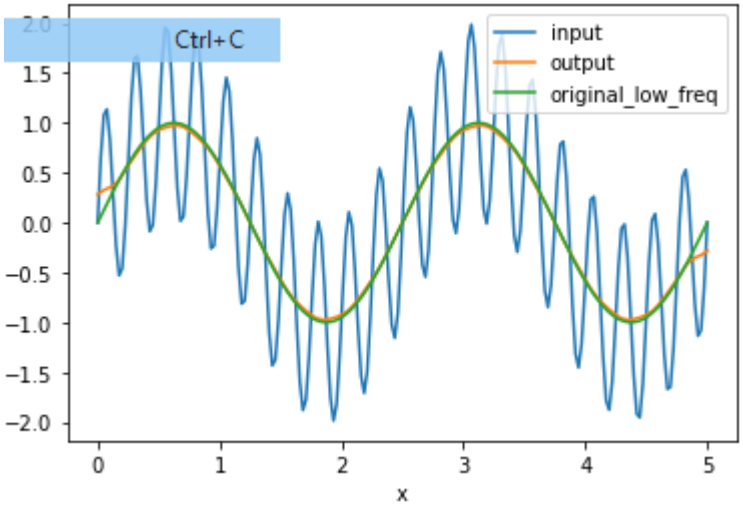
Part 1: 1D Filter

<insert visualization of the low-pass filter from proj1.ipynb here>

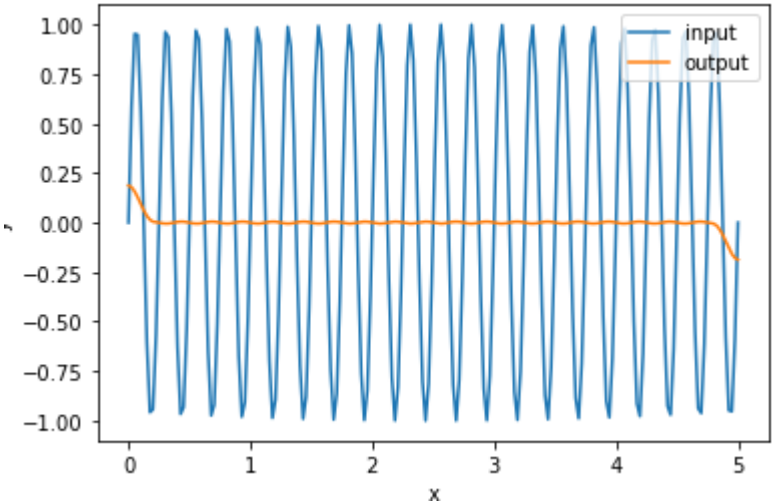


Part 1: 1D Filter

<insert visualization of filtered combined signal from proj1.ipynb here>

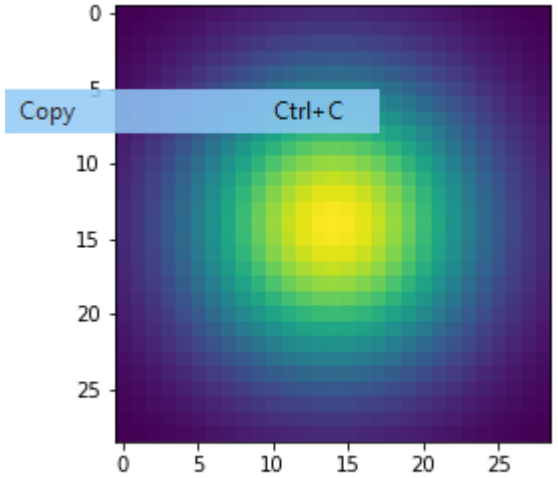


<insert visualization of the low-pass filter from proj1.ipynb here>



Part 2: Image Filtering

<insert visualization of the 2D Gaussian kernel from proj1.ipynb here>



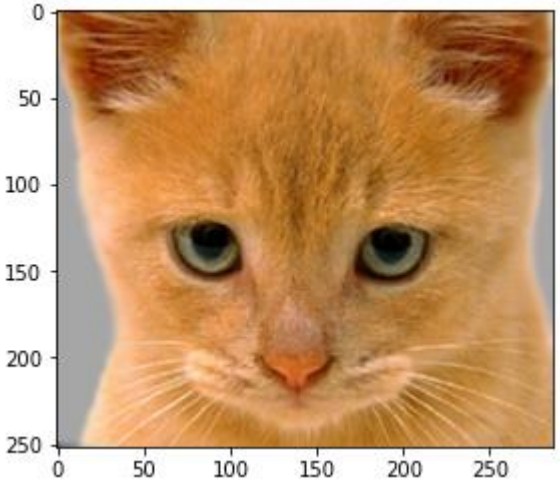
<Describe your implementation of my_imfilter() in words.>

I calculated each cell of the filtered image separately. I padded the signal with zeros so avoid out of bounds errors. Each cell was calculated by a dot product of the filter and a subset selection of the padded signal.

Part 2: Image filtering

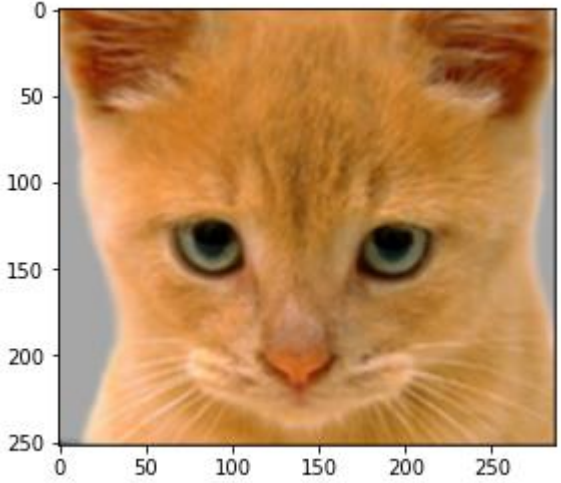
Identity filter

<insert the results from proj1_test_filtering.ipynb using 1b_cat.bmp with the identity filter here>



Small blur with a box filter

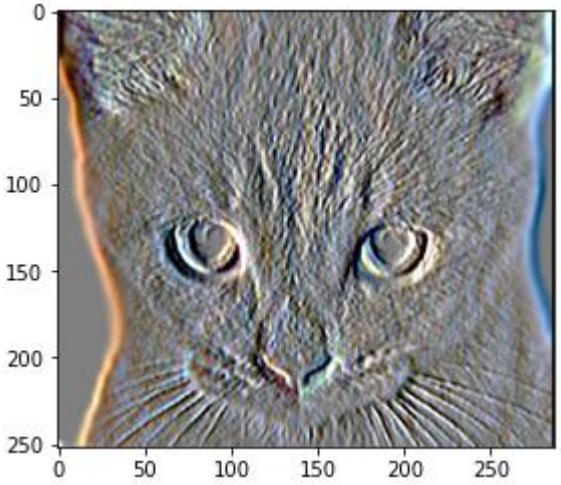
<insert the results from proj1_test_filtering.ipynb using 1b_cat.bmp with the box filter here>



Part 2: Image filtering

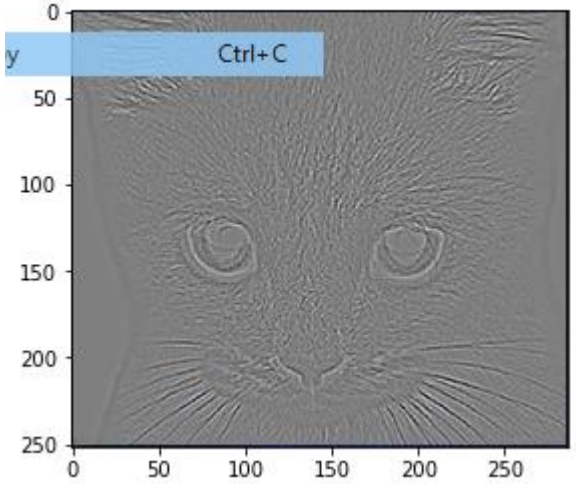
Sobel filter

<insert the results from proj1_test_filtering.ipynb using 1b_cat.bmp with the Sobel filter here>



Discrete Laplacian filter

<insert the results from proj1_test_filtering.ipynb using 1b_cat.bmp with the discrete Laplacian filter here>

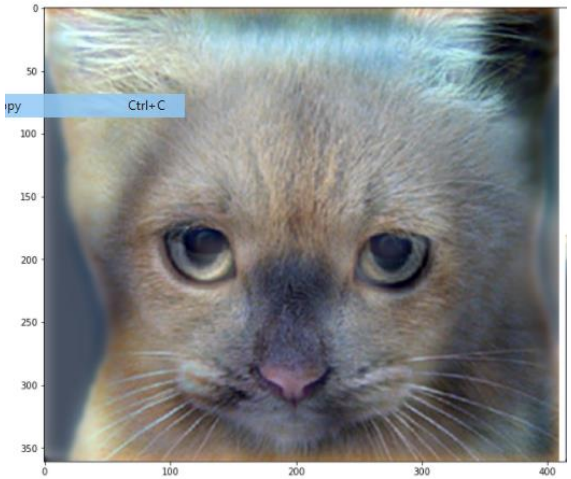


Part 2: Hybrid images manually using Pytorch

<Describe your implementation of create_hybrid_image() here.>

I used the 2d gaussian kernel to make low pass filters of both image1 and image2. Then, I subtracted the low frequency image2 from image2 to get the high frequency image2. Lastly, I added the low frequency image1 to high frequency image2

Cat + Dog



Cutoff frequency: 7

Part 2: Hybrid images manually using Pytorch

Motorcycle + Bicycle

Cutoff frequency: 7

Plane + Bird

re>

Cutoff frequency: 7

Part 2: Hybrid images manually using Pytorch

Einstein + Marilyn

<insert your hybrid image here>

Cutoff frequency: 7

Submarine + Fish


here>

Cutoff frequency: 7

Part 3: Hybrid images with PyTorch operators

Cat + Dog

Motorcycle + Bicycle

Part 3: Hybrid images with PyTorch operators

Plane + Bird

Einstein + Marilyn

re>

Part 3: Hybrid images with PyTorch operators

Submarine + Fish

Part 2 vs. Part 3

<Compare the run-times of Parts 2 and 3 here, as calculated in proj1.ipynb. What can you say about the two methods?>

Part 2 took 36 seconds. Part 3 took 3 seconds. Using built in Pytorch operations was much faster.

Tests

```
(proj1) PS C:\Users\Karan Sarkar\google drive\gt\computer vision\proj1_release> pytest
============================================= test session starts =============================================
platform win32 -- Python 3.6.12, pytest-6.0.1, py-1.9.0, pluggy-0.13.1
rootdir: C:\Users\Karan Sarkar\google drive\gt\computer vision\proj1_release
collected 13 items

proj1_code\proj1_unit_tests\test_2d.py .....                                                          [ 46%]
proj1_code\proj1_unit_tests\test_create_1D_Gaussian_kernel.py ...                                     [ 69%]
proj1_code\proj1_unit_tests\test_dft.py ...                                                           [ 84%]
proj1_code\proj1_unit_tests\test_my_1dfilter.py ..                                                    [100%]

============================================================================================== warnings summary ===============================================================================================
C:\Users\Karan Sarkar\miniconda3\envs\proj1\lib\site-packages\torchvision\io\video.py:2
  C:\Users\Karan Sarkar\miniconda3\envs\proj1\lib\site-packages\torchvision\io\video.py:2: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses
    import imp

-- Docs: https://docs.pytest.org/en/stable/warnings.html
=================================================================================== 13 passed, 1 warning in 21.90s ====================================================================================
(proj1) PS C:\Users\Karan Sarkar\google drive\gt\computer vision\proj1_release>
(proj1) PS C:\Users\Karan Sarkar\google drive\gt\computer vision\proj1_release>
(proj1) PS C:\Users\Karan Sarkar\google drive\gt\computer vision\proj1_release>
(proj1) PS C:\Users\Karan Sarkar\google drive\gt\computer vision\proj1_release>
```

Copy Ctrl+C

ests` on your final code

Pairs: This question must be answered individually

Conclusions

I learned how allowing more low frequency of an image through creates the general outline of an image. The high frequency creates the detail.

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!