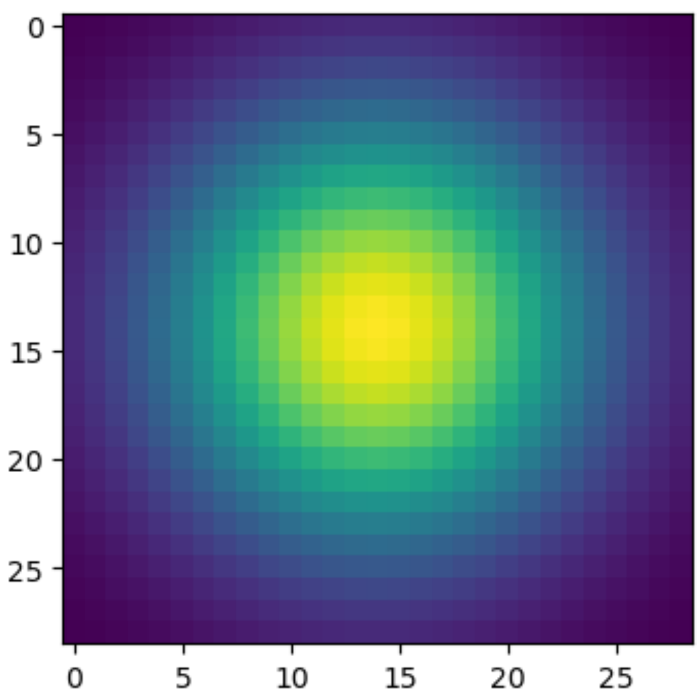
CS 6476 Project 1

Hardik Goel
hgoel7@gatech.edu
hgoel7
903536536

Part 1: Image filtering



my_conv2d_numpy()

For my implementation of conv2d I start with initiating the output filtered image with dimensions (m, n, c) same as input image.

We find the desired pad size for output image made from convolution image with filter. Using the padding size we pad our input image and loop it over to run filter and produce filtered image.

Part 1: Image filtering

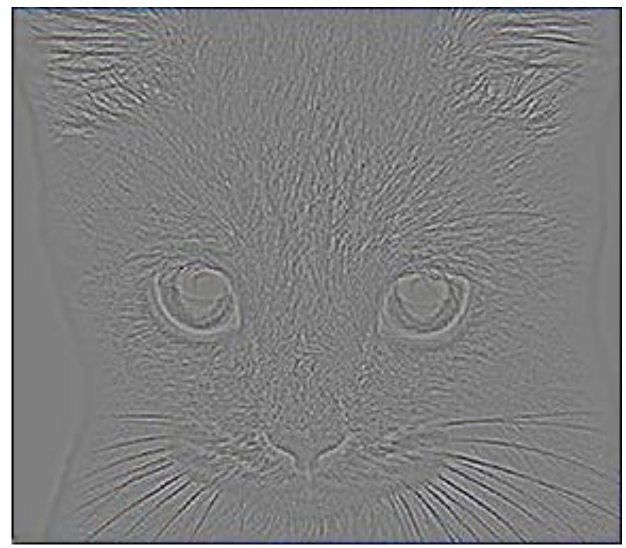
Identity filter

Small blur with a box filter

Part 1: Image filtering

Sobel filter

Discrete Laplacian filter



Part 1: Hybrid images

Create hybrid images has 3 main steps:

1. Find low frequencies of image1
2. Find high frequencies of image2
3. Clipping low and high frequencies together to generate a new hybrid image

Cat + Dog

Cutoff frequency: 7

Part 1: Hybrid images

Motorcycle + Bicycle

Cutoff frequency: 7

Plane + Bird

Cutoff frequency: 7

Part 1: Hybrid images

Einstein + Marilyn

Cutoff frequency: 13

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

Runtime of Part 2 was WAY faster than part 1. Part 1 took about 14 seconds to generate while Part 2 took 3 seconds, which is a huge improvement.

Part 3: Understanding input/output shapes in PyTorch

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 =>
Input: (410, 361, 1) Output: (408, 359, 3)
Stride = 2, padding = 0 =>
Input: (410, 361, 1) Output: (204, 180, 3)
Stride = 1, padding = 1 =>
Input: (410, 361, 1) Output: (410, 361, 3)
Stride = 2, padding = 1 =>
Input: (410, 361, 1) Output: (205, 181, 3)

Part 3: Understanding input/output shapes in PyTorch

How many filters did we apply to the dog image?

4

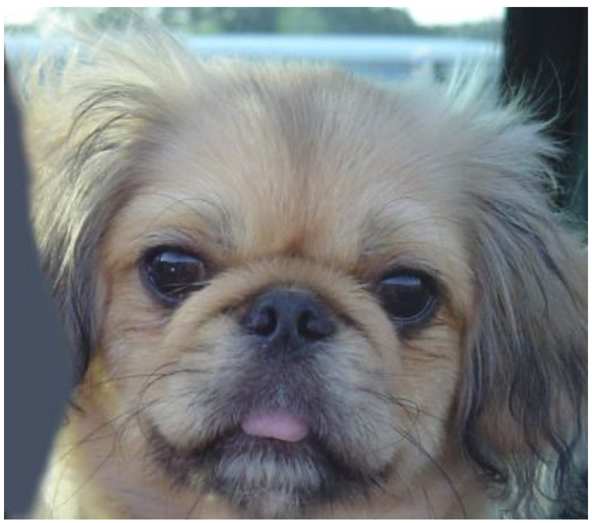
Section 3 of the handout gives equations to calculate output dimensions given filter size, stride, and padding. What is the intuition behind this equation?

The equation works because:

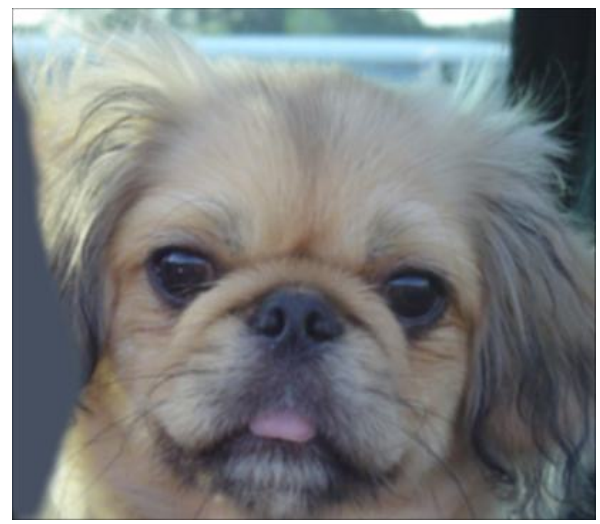
1. (+2p) part of the equation adjusts for padding on either sides of the image making it symmetric
2. (1/s) part of the equation represents stride i.e. the no. pixels the filter moves over. Higher stride results in lower resolution and vice versa. The 1/s represents that it's inversely proportional.

Part 3: Understanding input/output shapes in PyTorch

[insert visualization 0 here]



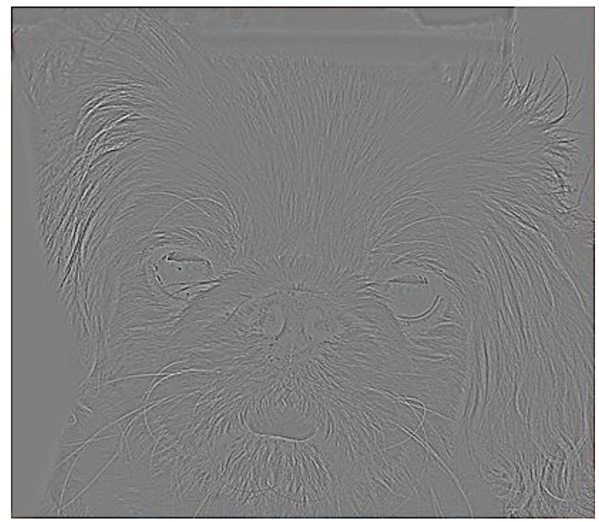
[insert visualization 1 here]



Part 3: Understanding input/output shapes in PyTorch

[insert visualization 2 here]

[insert visualization 3 here]



Part 4: Frequency Domain Convolutions

[Insert the visualizations of the dog image in the spatial and frequency domain]

[Insert the visualizations of the blurred dog image in the spatial and frequency domain]

Part 4: Frequency Domain Convolutions

[Insert the visualizations of the 2D Gaussian in the spatial and frequency domain]

[Why does our frequency domain representation of a Gaussian not look like a Gaussian itself? How could we adjust the kernel to make these look more similar?]

Part 4: Frequency Domain Convolutions

[Briefly explain the Convolution Theorem and why this is related to deconvolution]

Part 4: Frequency Domain Convolutions

[Insert the visualizations of the mystery image in the spatial and frequency domain]

[Insert the visualizations of the mystery kernel in the spatial and frequency domain]

Part 4: Frequency Domain Convolutions

[Insert the de-blurred mystery image and its visualizations in the spatial and frequency domain]

[Insert the de-blurred mystery image and its visualizations in the spatial and frequency domain after adding salt and pepper noise]

Part 4: Frequency Domain Convolutions

[What factors limit the potential uses of deconvolution in the real world? Give two possible factors]

[We performed two convolutions of the dog image with the same Gaussian (one in the spatial domain, one in the frequency domain). How do the two compare, and why might they be different?]

Conclusion

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