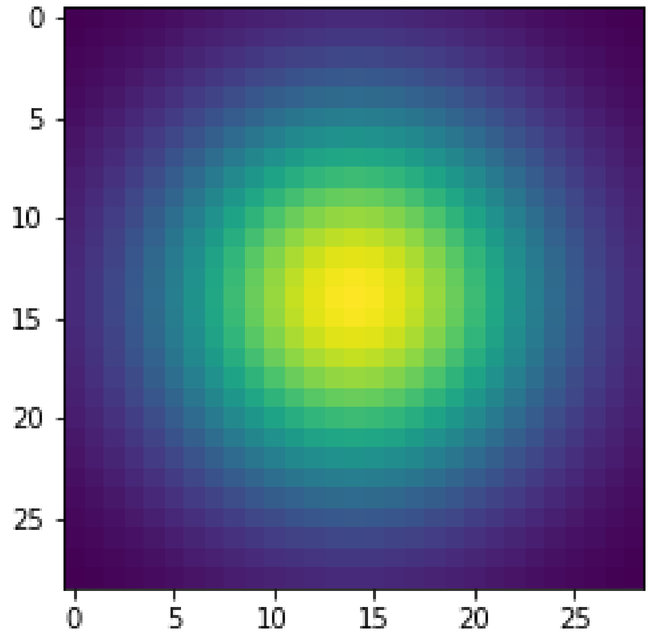
CS 6476 Project 1

[Manan Patel]
[mpatel608@gatech.edu]
[mpate608]
[903748003]

Part 1: Image filtering



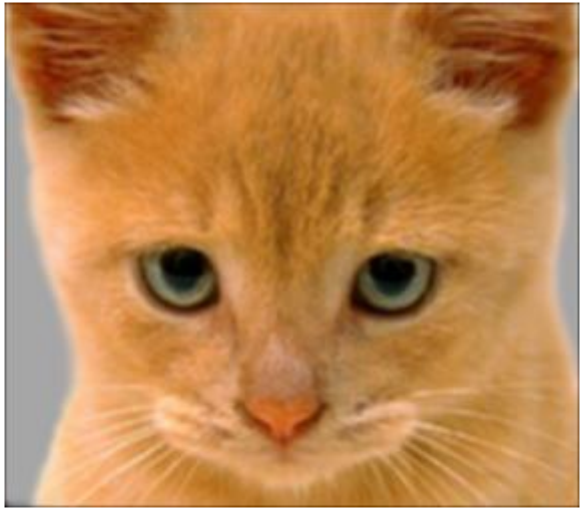
1. First, a 1D gaussian kernel was created given n and sigma. Here, n / 2 would be the mean and n would be the total size of the array and each index location would be filled with its probability density using gaussian distribution formula.

2. Then, to generate a 2D Gaussian, we use the property that it is nothing but the outer product of two 1D Gaussians.

Part 1: Image filtering

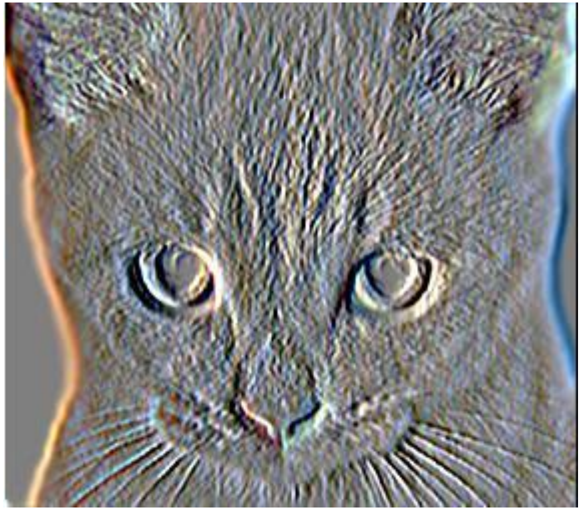
Identity filter

Small blur with a box filter

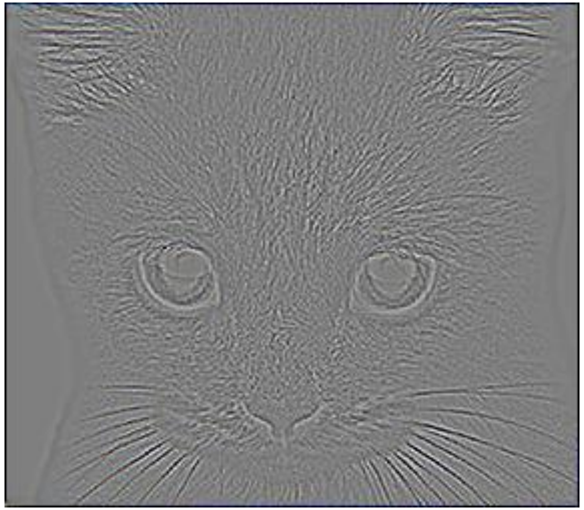


Part 1: Image filtering

Sobel filter



Discrete Laplacian filter

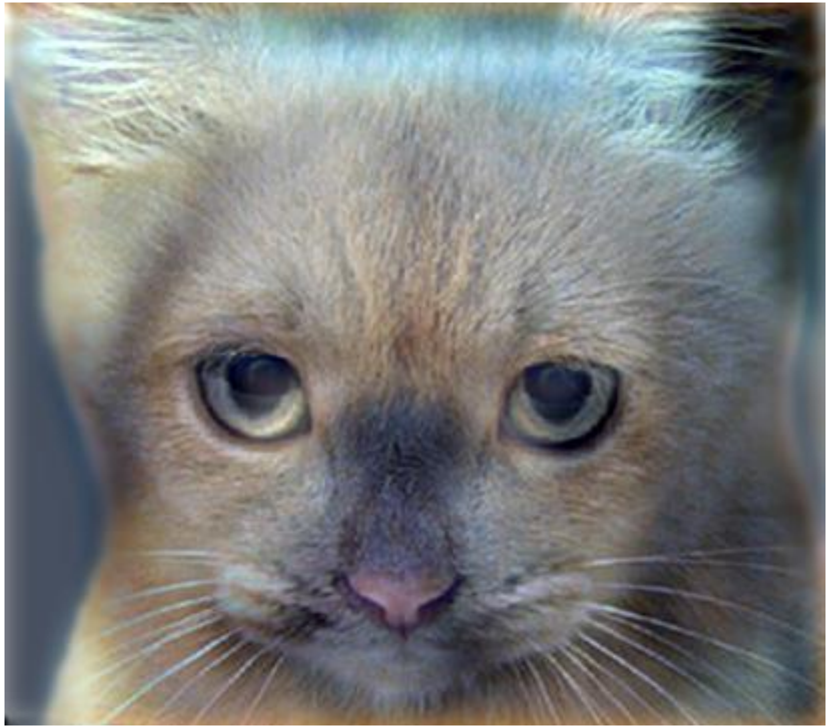


Part 1: Hybrid images

1. First based on a cut-off frequency, a 2D gaussian filter is generated. Using this filter, the low frequency image is generated by convolving the filter with it.

2. The high frequency image is generated by convolving the second image with the low frequency and then subtracting the output from the original image.

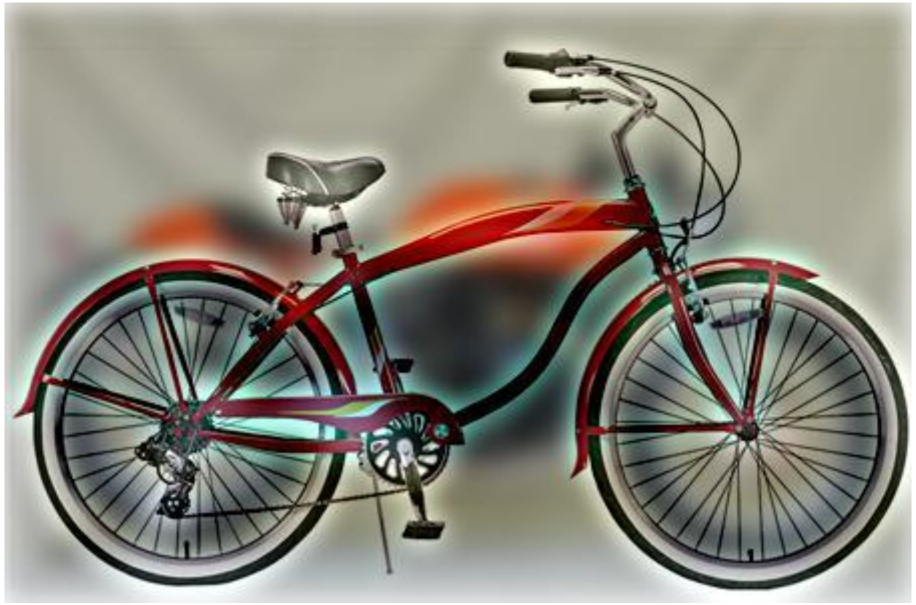
3. Now, simply adding the above two images and clipping it will give a hybrid image.



Cutoff frequency: 7

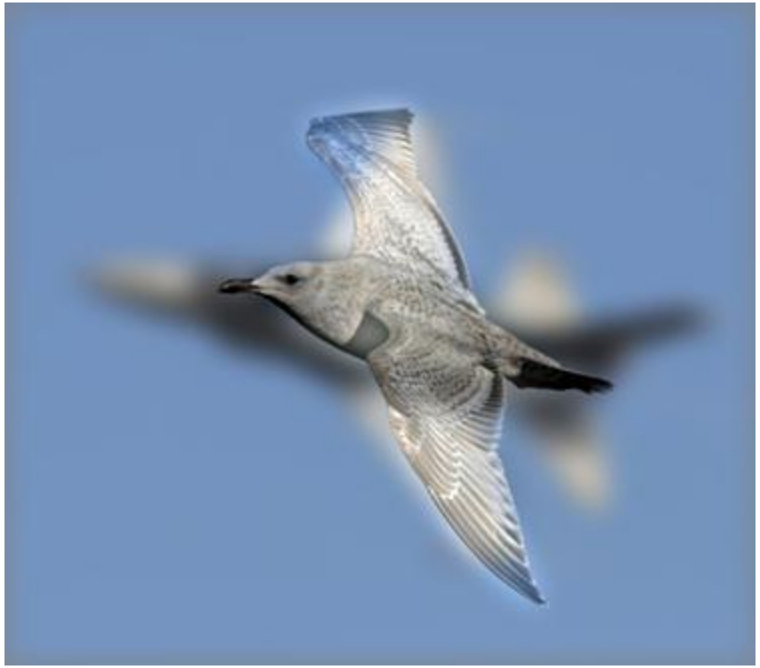
Part 1: Hybrid images

Motorcycle + Bicycle



Cutoff frequency: 8

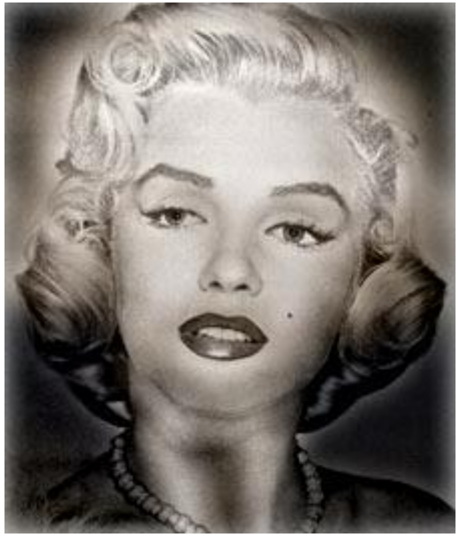
Plane + Bird



Cutoff frequency: 5

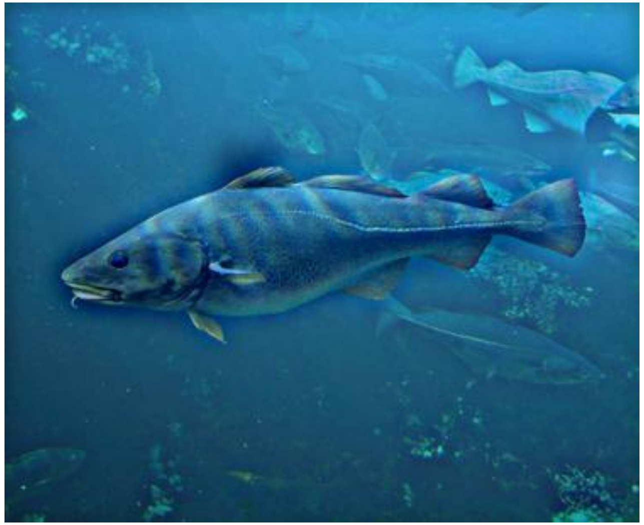
Part 1: Hybrid images

Einstein + Marilyn



Cutoff frequency: 4

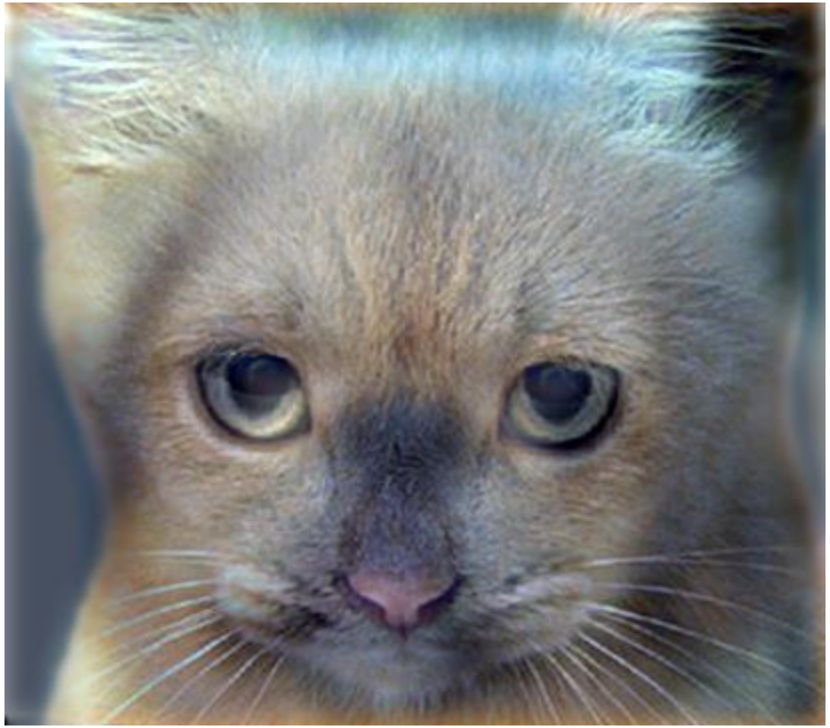
Submarine + Fish



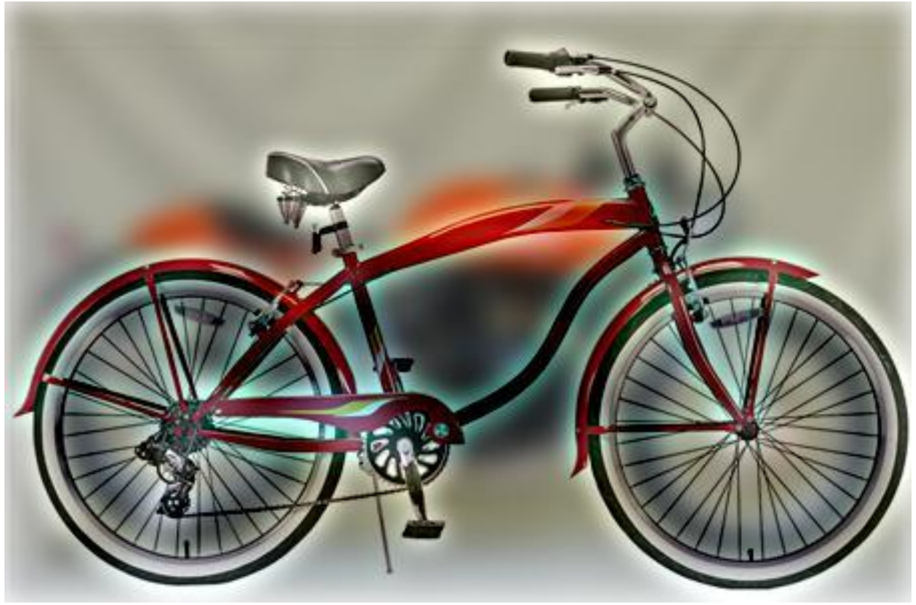
Cutoff frequency: 7

Part 2: Hybrid images with PyTorch

Cat + Dog

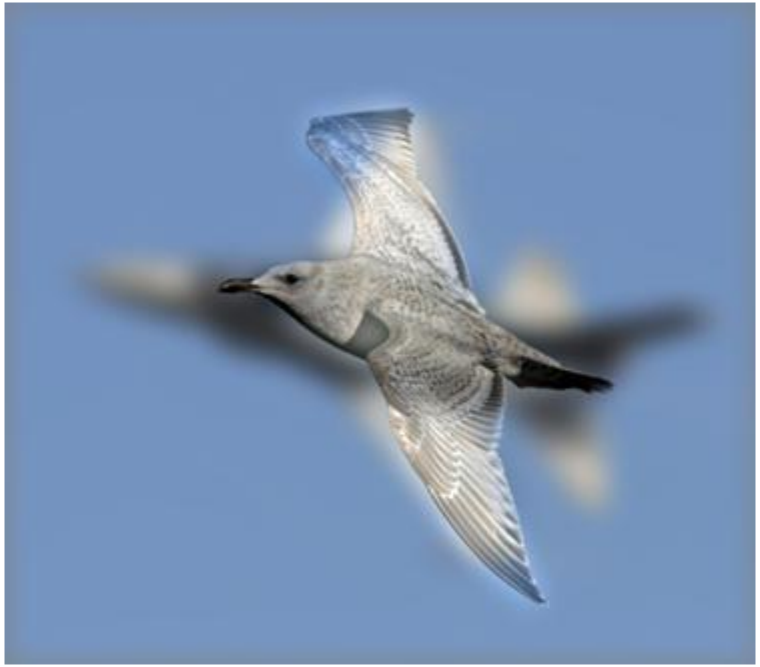


Motorcycle + Bicycle

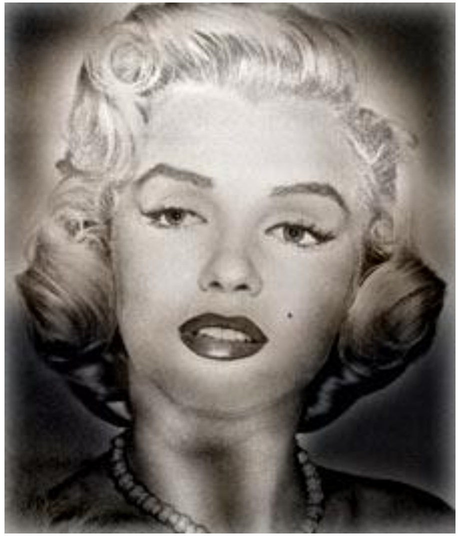


Part 2: Hybrid images with PyTorch

Plane + Bird

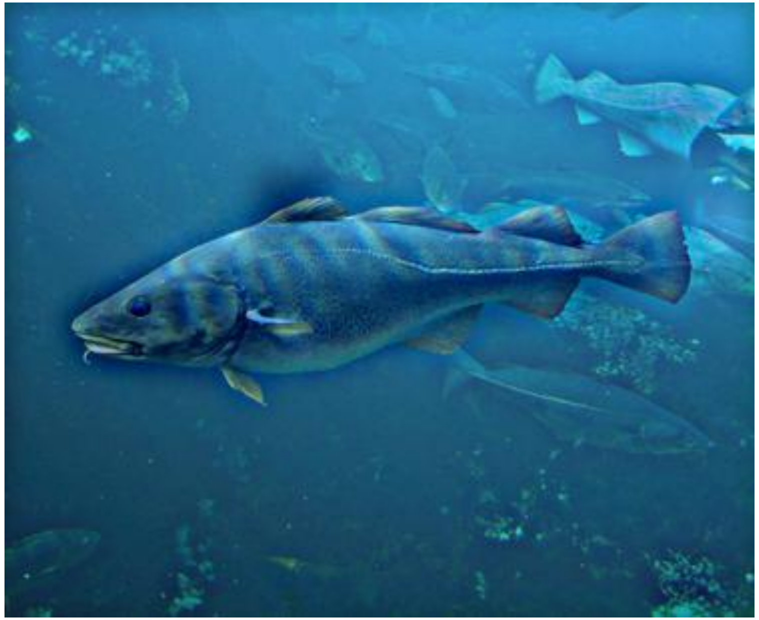


Einstein + Marilyn



Part 2: Hybrid images with PyTorch

Submarine + Fish



Part 1 vs. Part 2

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

Part 2 (2 sec) is faster than Part 1 (8 sec).

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?
Output_dim = (3 x 3)

Stride = 2, padding = 0?
Outupt_dim = (2 x 2)

Stride = 1, padding = 1?
Output_dim = (5x5)

Stride = 2, padding = 1?]
Output_dim = (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 → 408 x 359
Stride = 2, padding = 0 → 204 x 180
Stride = 1, padding = 1 → 410 x 361
Stride = 2, padding = 1?] → 205 x 181

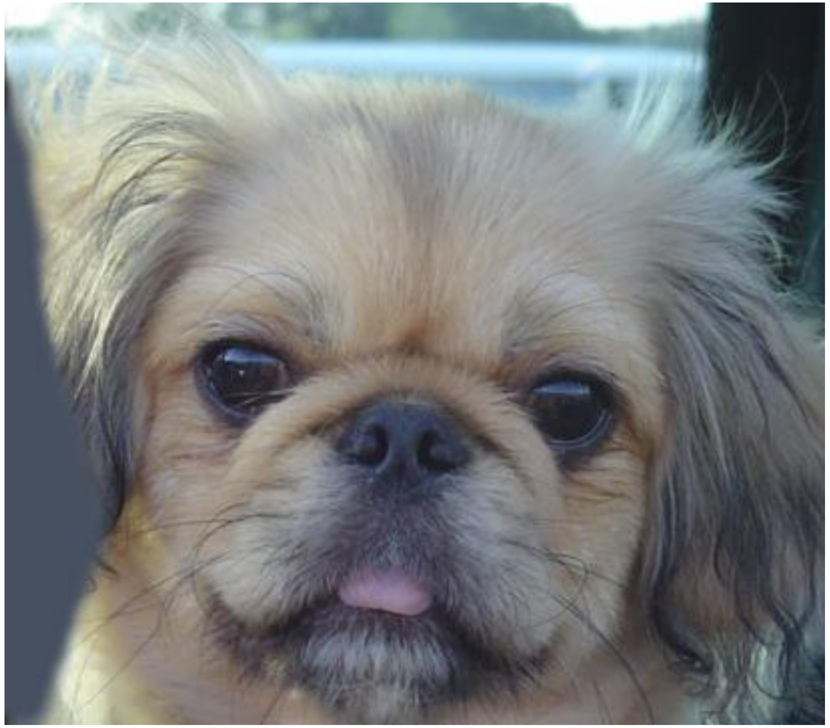
Part 3: Understanding input/output shapes in PyTorch

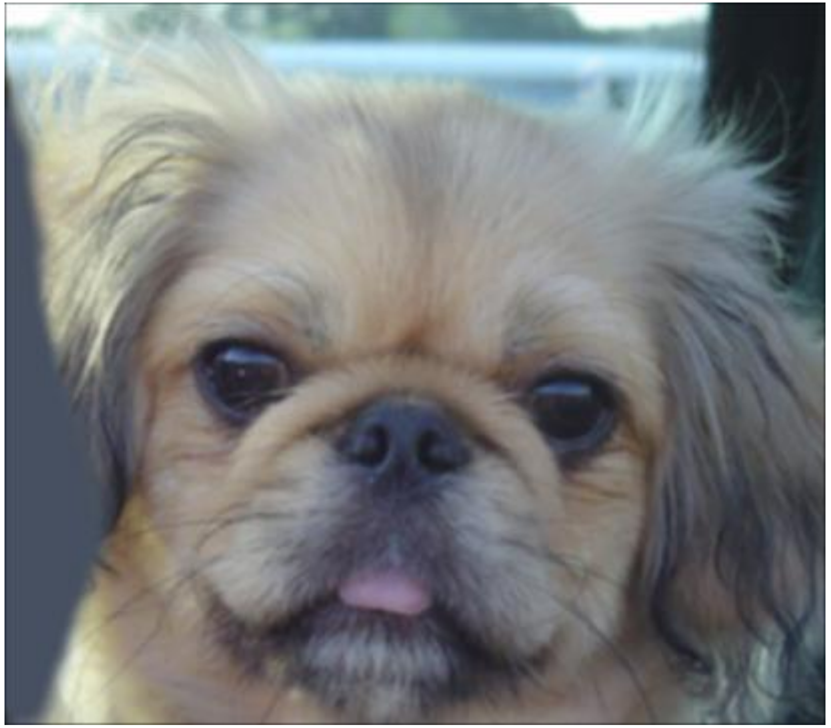
[How many filters did we apply to the dog image?]
4

[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 intuition comes basically from aligning the filter symmetrically whenever possible. In general, when an (m x n) filter is given, if m,n is odd, then padding on each side of row would be (m – 1)/ 2 and for columns (n – 1)/2.
For stride we can represent the pixel value calculated in an A. P. For example, if row size is m and m is odd, then the image reduces to a row size of
(m – 1) / 2 + 1. Similar idea can be applied to columns.

Part 3: Understanding input/output shapes in PyTorch





Part 3: Understanding input/output shapes in PyTorch

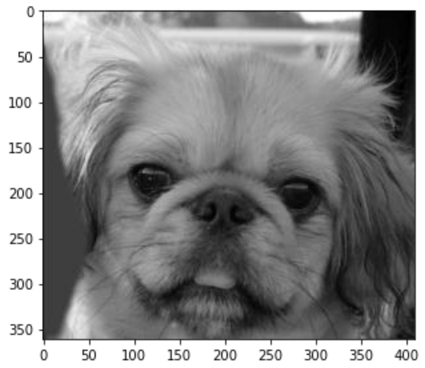
[insert visualization 2 here]

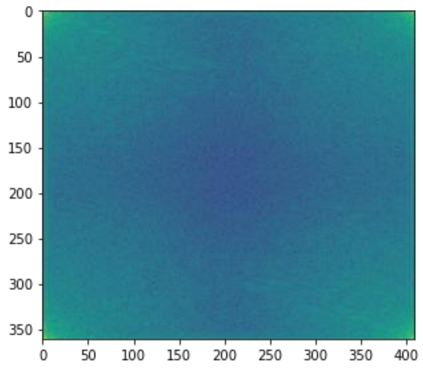
[insert visualization 3 here]

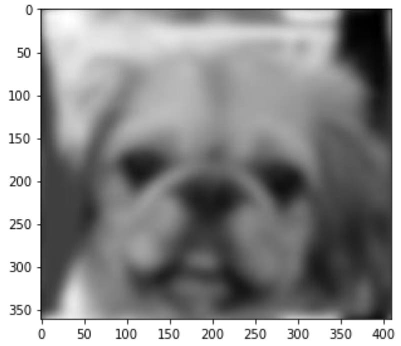
Part 4: Frequency Domain Convolutions

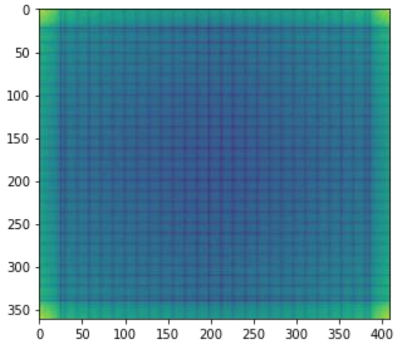
[Insert the visualizations of the dog image in the spatial and frequency domain]

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



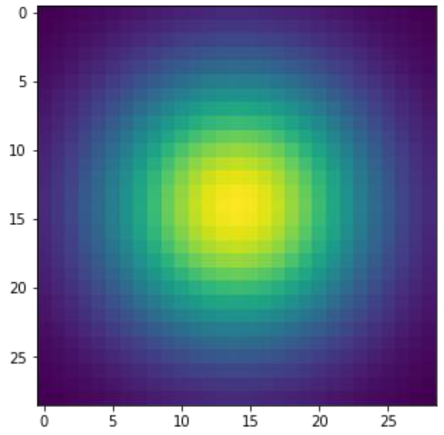


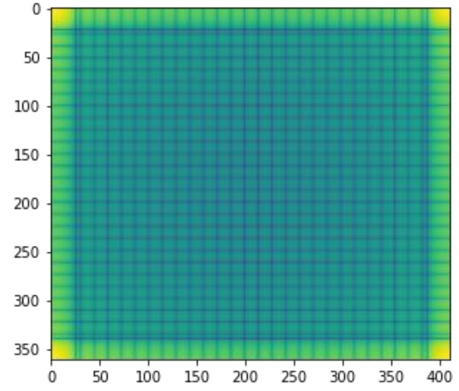




Part 4: Frequency Domain Convolutions

[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?]

- They don’t look similar because frequency information is represented differently.
- One possible way would be to increase the number of sequences in Fourier Transform.

Part 4: Frequency Domain Convolutions

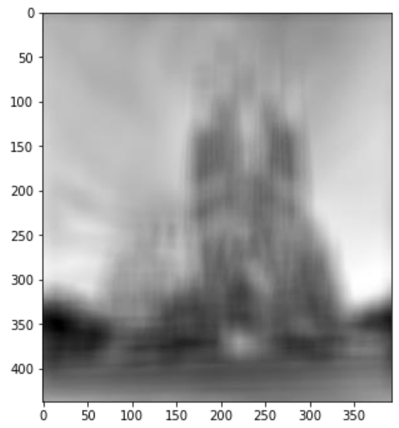
[Briefly explain the Convolution Theorem and why this is related to deconvolution]

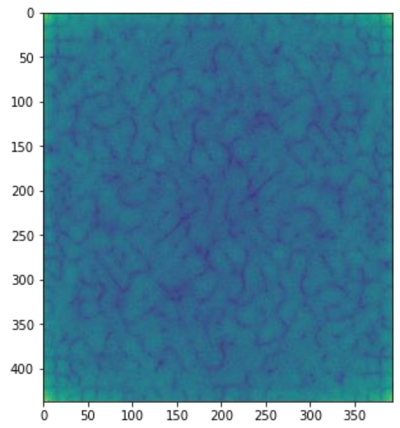
The convolution theorem states that, *convolution of two images in the spatial domain is equivalent to a dot product in the frequency domain.* This can be related to deconvolution as follows:

If we have information about a convolved image in the frequency domain and also the frequency domain representation of one of the images, then we can extract the second image by simply dividing the convolved image with image one and get the frequency representation of image two. Now, image 2 can be transformed to spatial domain to get the final results.

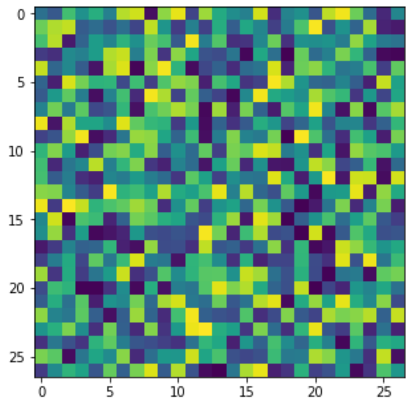
Part 4: Frequency Domain Convolutions

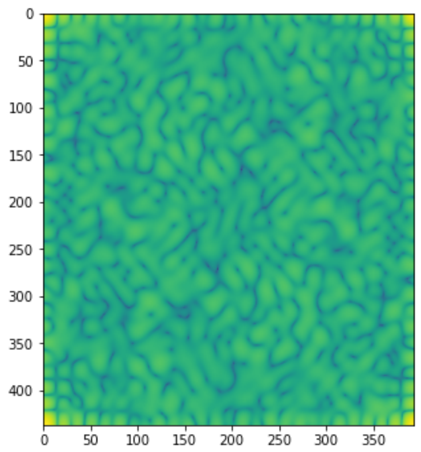
[Insert the visualizations of the mystery image in the spatial and frequency domain]





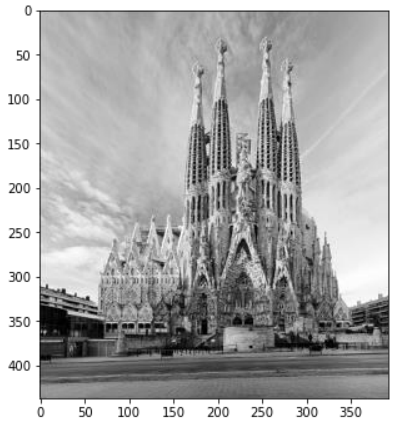
[Insert the visualizations of the mystery kernel in the spatial and frequency domain]

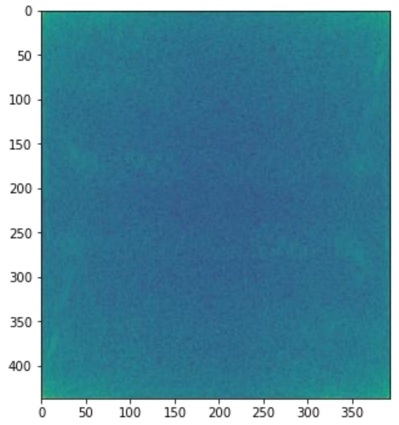




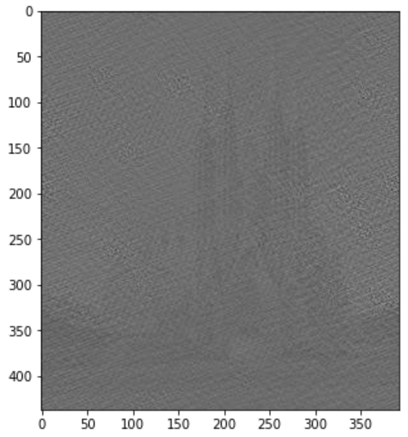
Part 4: Frequency Domain Convolutions

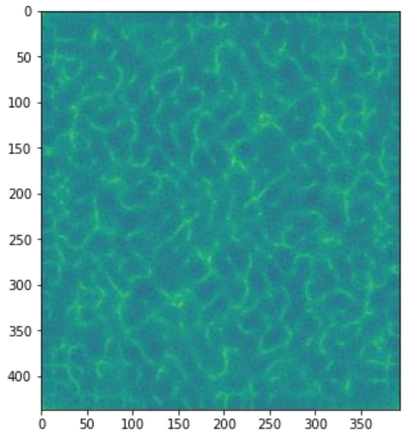
[Insert the de-blurred mystery image and its visualizations in the spatial and frequency domain]





[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]

- Dimly lit images: here we will encounter a problem of division with zero if the frequency domain representation is very close to zero.
- Scaling: When we convolve with a filter, the image is scaled and the final image may lose its information due to clipping.

[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?]

Convolution in the spatial domain is more detailed than in the frequency domain as there is a loss of information when we represent anything in terms of Fourier transform since it is complete only when we take infinite sequences.

Conclusion

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

The cut-off frequency determines the point after which all the other frequencies will be attenuated. Selecting the right cut-off frequency in an image depends mainly on its RGB value and size.

Whereas determining the correct combination for a hybrid image, one must analyse the human eye interpretation of both the images and verify if they are compatible or not.