Project 1: Image Filtering and Hybrid Images

Team 11

Sohaila Zaki 201-800-998
Ibrahim Ibrahim 201-800-739

Q1: Explicitly describe image convolution: the input, the transformation, and the output. Why is it useful for computer vision?

Convolution filtering is used to modify the spatial frequency characteristics of an image. A convolution is done by multiplying a pixel's and its neighboring pixels color value by a matrix.

The input is a 2D image and the kernel. A kernel is a small matrix of numbers that is used in image convolutions.
The transformation is used to describe the image using the basis of the kernel.
The output is a new modified filtered image.

Convolution is useful for computer vision as it is used to Smooth, Sharpen, Intensify, and Enhance Images.

Q2: What is the difference between convolution and correlation? Construct a scenario which produces a different output between both operations.

The main difference between convolution and correlation is that when we convolve an image with a filter, the filter should be flipped first. While correlation does not require applying flipping to the filter.

The convolution is the same as correlation when the kernel is Symmetric.

Symmetric Kernel = $\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$

But if the kernel is Non symmetric, correlation and convolution won't be the same.

Q3: What is the difference between a high pass filter and a low pass filter in how they are constructed, and what they do to the image? Please provide example kernels and output images.

Low pass filter: An example of the low pass filter is an averaging filter that takes the average of the neighboring pixels. The LPF is used to blur an image or smooth it.

High pass filter: the high pass filter highlights the edges by subtracting two pixels, so that the high values are highlighted.

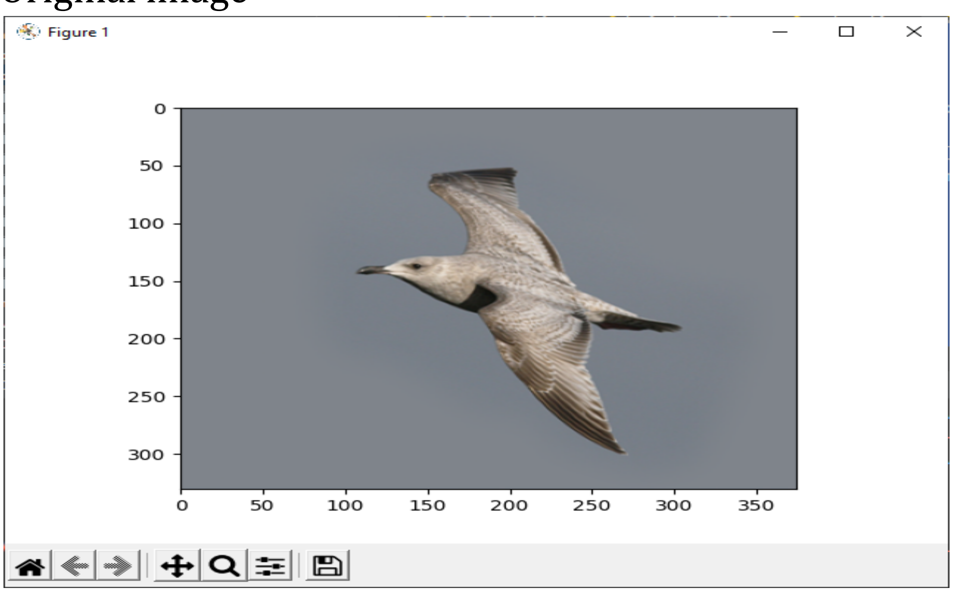
The HPF is used for edge detection.

Examples:

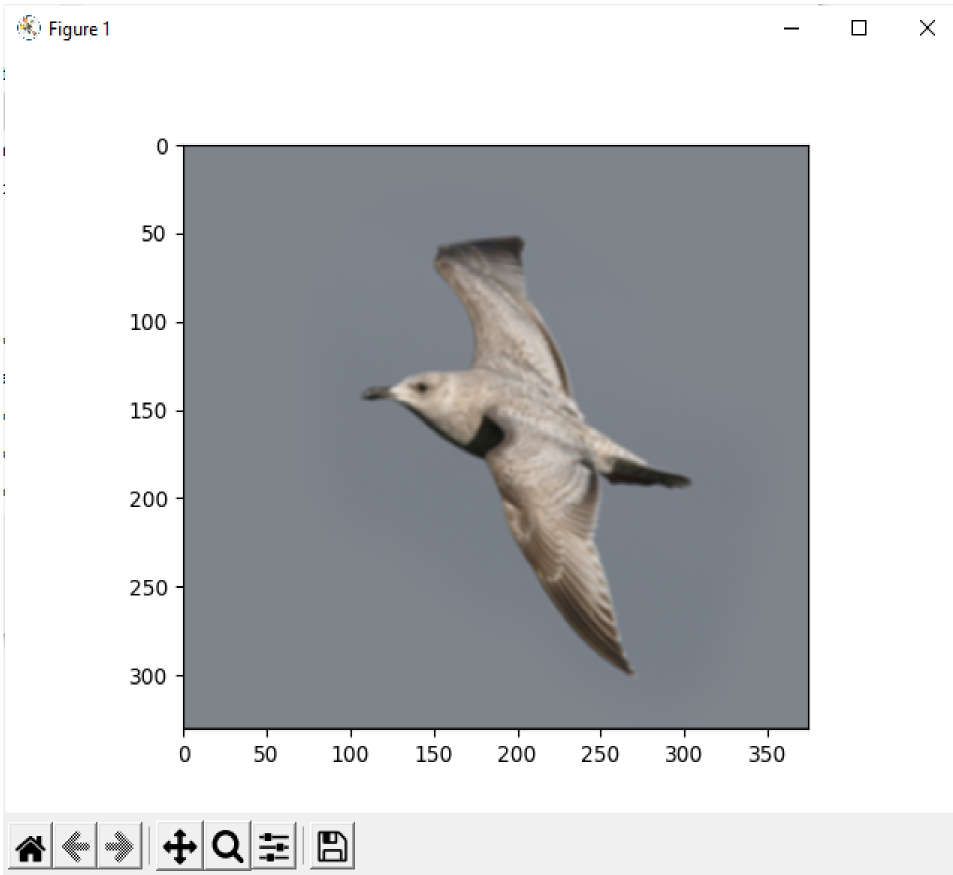
1) LPF

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Original image



Output image

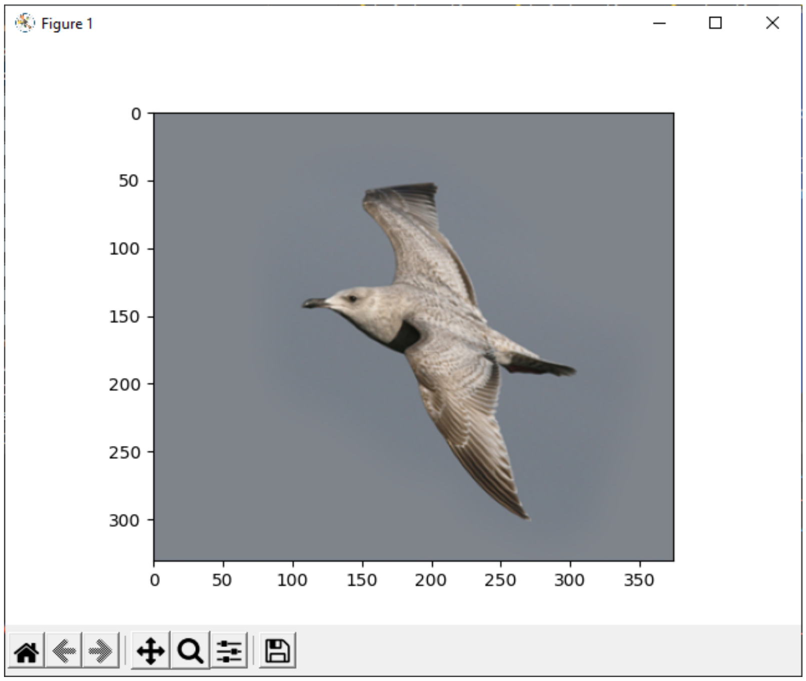


Output for gaussian filter

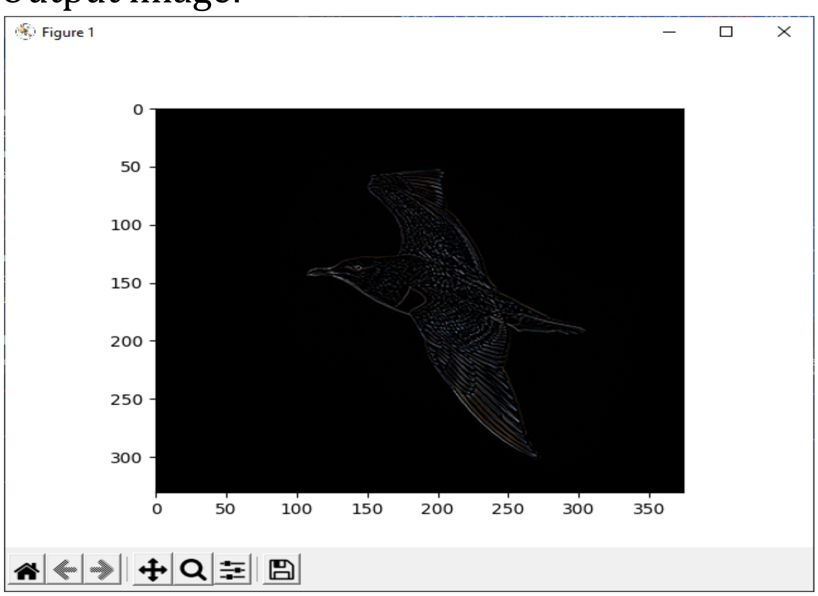
2) HPF (laplacian filter) =

0	1	0
1	-4	1
0	1	0

Original image:



Output image:



Q4: How does computation time vary with filter sizes from 3 × 3 to 15 × 15 (for all odd and square sizes), and with image sizes from 0.25 MPix to 8 MPix (choose your own intervals)? Measure both using scipy.ndimage.convolve or scipy.ndimage.correlate to produce a matrix of values. Use the skimage.transform module to vary the size of an image. Use an appropriate charting function to plot your matrix of results, such as Axes3D.scatter or Axes3D.plot surf ace.

Do the results match your expectation given the number of multiply and add operations in convolution?

The computation time increases as the size of the filter increase, since The complexity of 2D convolution is n^2 * m^2 multiplications and additions.

so if we have 8 MPix to 0.25 MPix image size then it would contribute to the most of the complexity. While changing the kernel size from 3*3 to 15*15 would also increase the number of computation done by a convolution but it would not be as significantly effective as changing in the image size