

Project 1 Questions

Instructions

- 4 questions.
- Write code where appropriate.
- Feel free to include images or equations.
- Please make this document anonymous.
- On upload, **Gradescope will ask you to assign question numbers to your pages.** Making each question end with a page break after your answer is a good way to ease this process.

Questions

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

A1: The convolution operation takes a filter F of size $p \times q$ and an image I of size $m \times n$. It then uses the filter F to, for each pixel in image I , compute a weighted sum of the surrounding pixels. Convolutions of image data take the form: $y(r, c) = \sum_{i,j} I(i, j)F(r - i, c - j)$

Convolutions allows us to enhance images (e.g. denoising, increase contrast, resizing, etc.), extract information from images (e.g. texture, edges, distinctive points, etc.) or detect patterns (e.g. template matching).

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

Please use [`scipy.ndimage.convolve`](#) and [`scipy.ndimage.correlate`](#) to experiment!

A2: The difference between the correlation operation and the convolution operation is that in the case of the latter the filter is flipped on the horizontal axis. In other words the filter is rotated by 180 degrees. If the filter matrix is symmetric across the horizontal axis there is no difference between the correlation and convolution operation. However, if this is not the case, there will be a difference in the produced output.

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.

A3: Your answer here.

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_surface*](#).

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

Image: [RISDance.jpg](#) (in the .tex directory).

A4: Your answer here.