1 Median Filtering

Recall that a convolutional kernel in image contexts involves calculating the sum of the elementwise products of the kernel with a window of pixels in the image, then sliding the kernel across the image, repeating the process for every pixel in the image.

(a) Consider the following 3x3 kernel:

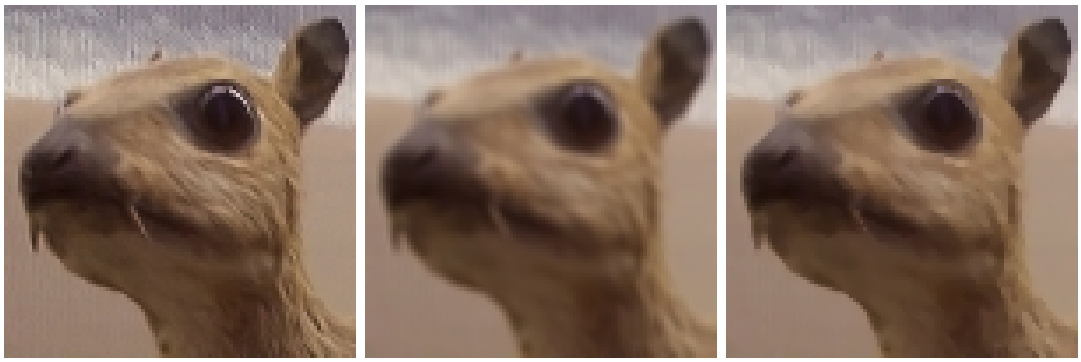
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

This is often referred to as a "box-blur" kernel. Why does it have a blurring effect?

The box-blur kernel effectively averages each pixel with its neighbors. This means that the resulting image will be blurred, because sharp edges will become somewhat gradiented.

(b) Now consider a related technique, known as *median filtering*. Here, for a given window of pixels, the center pixel is replaced with the median of the pixels in the window. This process is then repeated across the entire image, just as is done with convolution.

For your curiosity, depicted below are the results of applying 3x3 filters to an image (original left, box-blur center, median right):



Can you think of a scenario where median filtering would be preferable to box-blur filtering? What about the other way around?

Median filtering can be very useful for denoising, where the noise is extremely pronounced, because it eliminates outliers, whereas box-blur filtering will not. Box-blur filtering is useful for universally blurring images, and is also easier to compute.

(c) For ConvNets, are median filters similar to convolutional layers? Are they still useful? Why or why not?

Median filters are not similar to convolutional layers, because they do not learn weights. In this sense, they are more similar to pooling layers. However, they can still be useful, because they can be used to denoise images.

(d) Let $K \times K$ denote filter size, P padding size, S stride length, and suppose we are operating on an image of size $H \times W$. What is the runtime complexity of convolution? How about median filtering? Assume no results are cached across different pixels.

Hint: You can find the median of n numbers in $O(n)$ time.

The image after padding is effectively $(H+2P) \times (W+2P)$. Stride reduces computation by approximately $1/S^2$, because each $S \times S$ window has only one elementwise product operation. For each pixel considered, there are K^2 multiplications. Thus, the complexity of convolution is $O(\frac{K^2}{S^2}(H+P)(W+P))$, which, with the assumption that padding is always less than the size of the image, is $O(\frac{K^2HW}{S^2})$. The complexity of median filtering is the same.

2 Cross-Entropy, Revisited

hello