Correlation vs Convolution

Well, both are image-processing techniques that are used to extract information from an image array.

But the main difference is that the kernel, or the filter is rotated by 180 degrees while performing the sum of elementwise dot product between the kernel and the image array.

If the kernels are symmetric, then both processes would be similar as rotating will not change the kernel in any way.

In the code that Is provided along with the pdf, one can see that the filter is flipped both horizontally and vertically for correlation before convolving over the image.

```
flipped_kernel = np.flipud(np.fliplr(kernel))
```

If the filter for 1D convolution is (3,7,5), in correlation, it would turn into (5, 7,3).

If one goes deeper, it is easy to understand that convolution is associative. That is, the order in which the convolutions are applied does not affect the result.

For instance, if F1 and F2 are two filers convolved over image I, then (F1 * F2)* I = F1 * (F2 * I). I can combine two filters and then convolve over an image to get the same result as convolving with a single filter one after the other.

Usually, convolution is used for processes such as smoothing [think Gaussian] whereas correlation is used for template matching to an image.

Reference : Microsoft Word - Correlation and Convolution.doc (umd.edu)