Calculation

* In our image processing problem we will use a method of applying a kernel or convolution matrix to de-noise an image. This is accomplished by means of convolution between a kernel and an image to achieve the effects such as blurring, sharpening, embossing, edge detection, and more.

What is Convolution?

* Convolution is a general purpose filter effect for images.
* It works by determining the value of a central pixel by adding the weighted values of all its neighbors together.
* The output is the new modified filtered image.

The Process of a Convolution

* A convolution is done by multiplying a pixel and its neighboring pixels color values by a kernel.
* A kernel is usually a small matrix of numbers that is used in image convolutions.
* Differently sized kernels containing different patterns of numbers produce different results under convolution.

|  |  |  |
| --- | --- | --- |
| Identity | {\displaystyle {\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}}} | [Vd-Orig.png](https://en.wikipedia.org/wiki/File:Vd-Orig.png) |
| Edge Detection | {\displaystyle {\begin{bmatrix}\ \ 1&0&-1\\\ \ 0&0&\ \ 0\\-1&0&\ \ 1\end{bmatrix}}} | [Vd-Edge1.png](https://en.wikipedia.org/wiki/File:Vd-Edge1.png) |
| {\displaystyle {\begin{bmatrix}0&\ \ 1&0\\1&-4&1\\0&\ \ 1&0\end{bmatrix}}} | [Vd-Edge2.png](https://en.wikipedia.org/wiki/File:Vd-Edge2.png) |
| {\displaystyle {\begin{bmatrix}-1&-1&-1\\-1&\ \ 8&-1\\-1&-1&-1\end{bmatrix}}} | [Vd-Edge3.png](https://en.wikipedia.org/wiki/File:Vd-Edge3.png) |
| **Sharpen** | {\displaystyle {\begin{bmatrix}\ \ 0&-1&\ \ 0\\-1&\ \ 5&-1\\\ \ 0&-1&\ \ 0\end{bmatrix}}} | [Vd-Sharp.png](https://en.wikipedia.org/wiki/File:Vd-Sharp.png) |
| **Box Blur** (normalized) | {\displaystyle {\frac {1}{9}}{\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}}} | [Vd-Blur2.png](https://en.wikipedia.org/wiki/File:Vd-Blur2.png) |
| **Gaussian Blur** (approximation) | {\displaystyle {\frac {1}{16}}{\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}}} | [Vd-Blur1.png](https://en.wikipedia.org/wiki/File:Vd-Blur1.png) |

* The size of a kernel is arbitrary, we will use a 3x3 matrix.

Example of a kernel:

Let’s take worked example:

   

In this example we apply a convolution to the pixel value (201).

This is shown below:

New pixel value.

Divide by the sum of the kernel elements.

Convolution Formula

Where:

= the coefficient of a convolution kernel at position *i,j* (in the kernel)*.*

= the data value of the pixel that corresponds to

= the dimension of a kernel, assuming a square kernel (if = 3, the kernel is 3x3).

= either the sum of the coefficients of the kernel, or 1 if the sum of the coefficients is 0.

= the output pixel value.

(In cases where is less than 0, is clipped to 0).