## **Examples of 2D convolution**

How is the convolution operation used in practice?

1. Let us consider a 3x3 kernel and run it over an image, pixel-by-pixel.
2. This is done to re-estimate every pixel in that 3x3 neighborhood

|  |  |  |  |
| --- | --- | --- | --- |
| **Input 30 x 30** | **conv** | **Kernel 3x3** | **Output 30x30 (blur)** |
|  | \* | |  |  |  | | --- | --- | --- | | 1/9 | 1/9 | 1/9 | | 1/9 | 1/9 | 1/9 | | 1/9 | 1/9 | 1/9 | |  |

* 1. Here, we can see that the kernel is essentially an average operation, so what it does is it converts the value of every pixel to of its original value.
  2. In any photo editing tool like GIMP or Photoshop, when we select an image blur, we are essentially performing a convolution operation using an average valued kernel.

1. Let’s look at another convolution operation

|  |  |  |  |
| --- | --- | --- | --- |
| **Input 30 x 30** | **conv** | **Kernel 3x3** | **Output 30x30 (sharpens)** |
|  | \* | |  |  |  | | --- | --- | --- | | 0 | -1 | 0 | | -1 | 5 | -1 | | 0 | -1 | 0 | |  |

* 1. Here, the selected pixel is magnified by multiplying by 5 and then we subtract the 4 neighbors from it. This results in a sharper image, as it boosts the current pixel, thereby making it appear more prominent when compared to its neighbors.

1. Let’s look at one more example

|  |  |  |  |
| --- | --- | --- | --- |
| **Input 30 x 30** | **conv** | **Kernel 3x3** | **Output 30x30 (Edge detection)** |
|  | \* | |  |  |  | | --- | --- | --- | | 1 | 1 | 1 | | 1 | -8 | 1 | | 1 | 1 | 1 | |  |

* 1. Here, pixels near others pixels of the same value are reduced to 0, leaving only the edges.