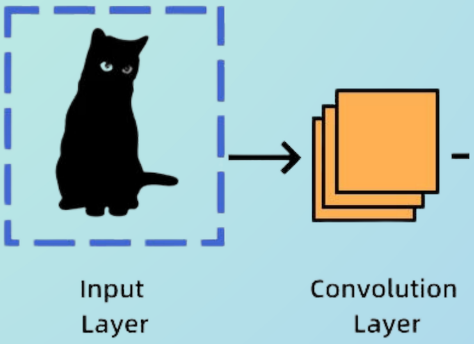
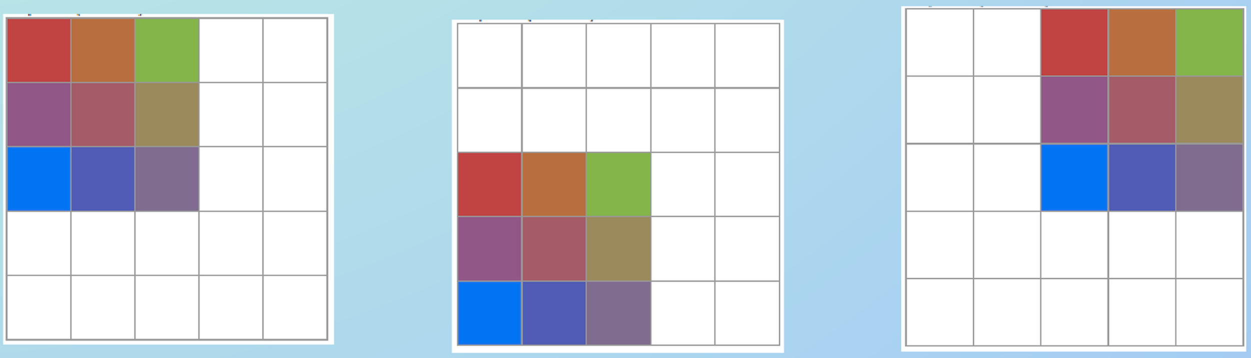
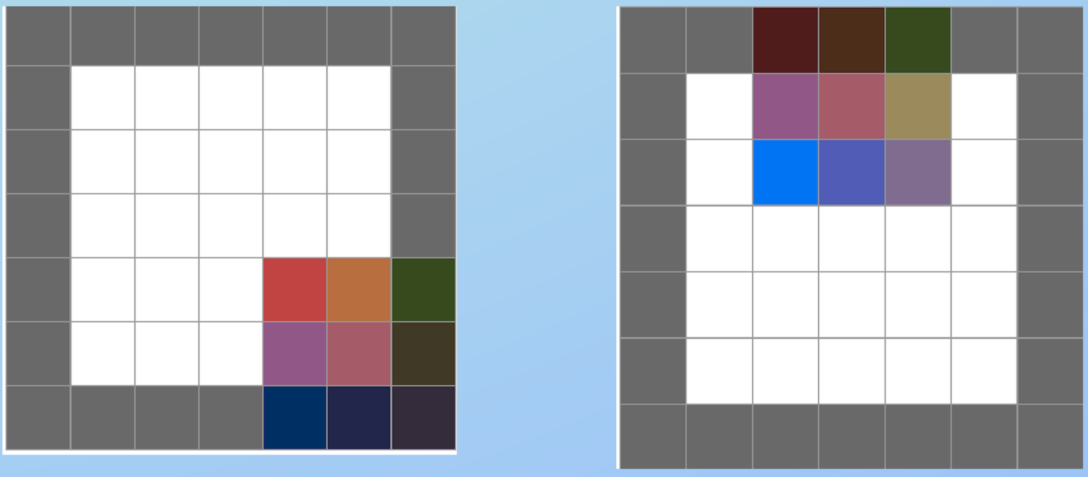
Fundamentals of Deep Learning

Convolutional Neural Networks : Stride and Padding







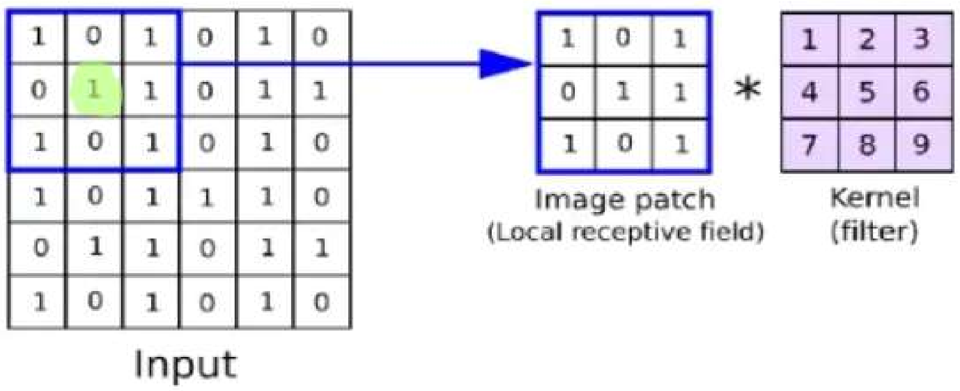
Shivang Kainthola

Kernels in Convolutional Neural Networks

Kernels are the key to detecting features in convolutional neural networks.

The kernel is centered on one pixel, and each weight in the kernel will be multiplied with the pixel's values, like a matrix multiplication.

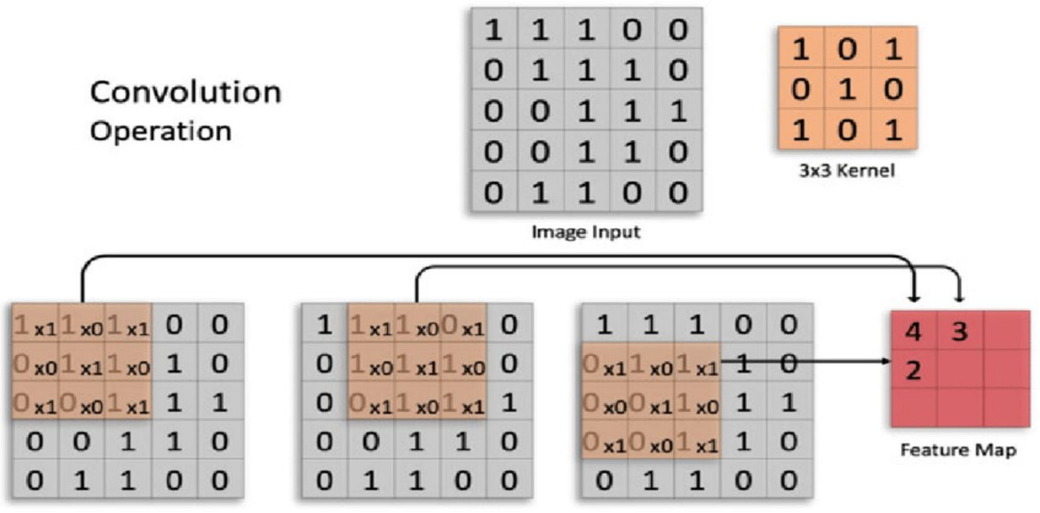
Image source : https://medium.com/@jayad1353/the-structure-of-convolutional-neural-networks-cnn-9d8c5c1c8aec



⟶ The process is called convolution, and the output is a feature map which is used by deeper convolution layers to detect complex features.

⟶ A kernel will cover the entire image, by moving over the entire image, pixel by pixel.

Image source : https://www.researchgate.net/figure/A-3-3-kernel-is-introduced-in-this-example-We-adopt-stride-of-1-ie-sliding-the_fig2_343356912



But how exactly do we decide the movement of a kernel ?

The Stride of a Kernel

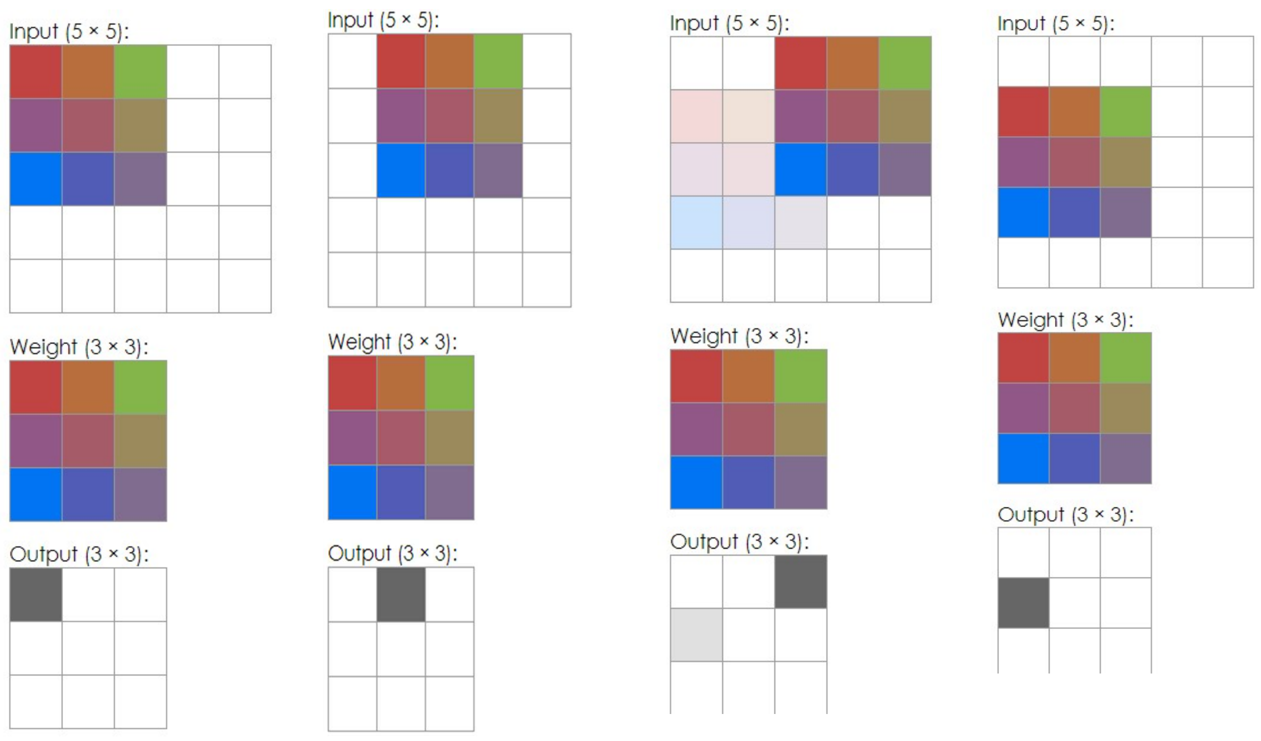
The stride of a kernel is the number of steps the kernel takes in one movement across the image. By default, the step size is one step- i.e. the kernel moves over by one pixel at time.

⟶ The most common movement pattern is from left to right across each row, then down to the next row and repeating until the entire image is covered.

⟶ The movement of a kernel can only be vertical and horizontal, there are no diagonal movements.

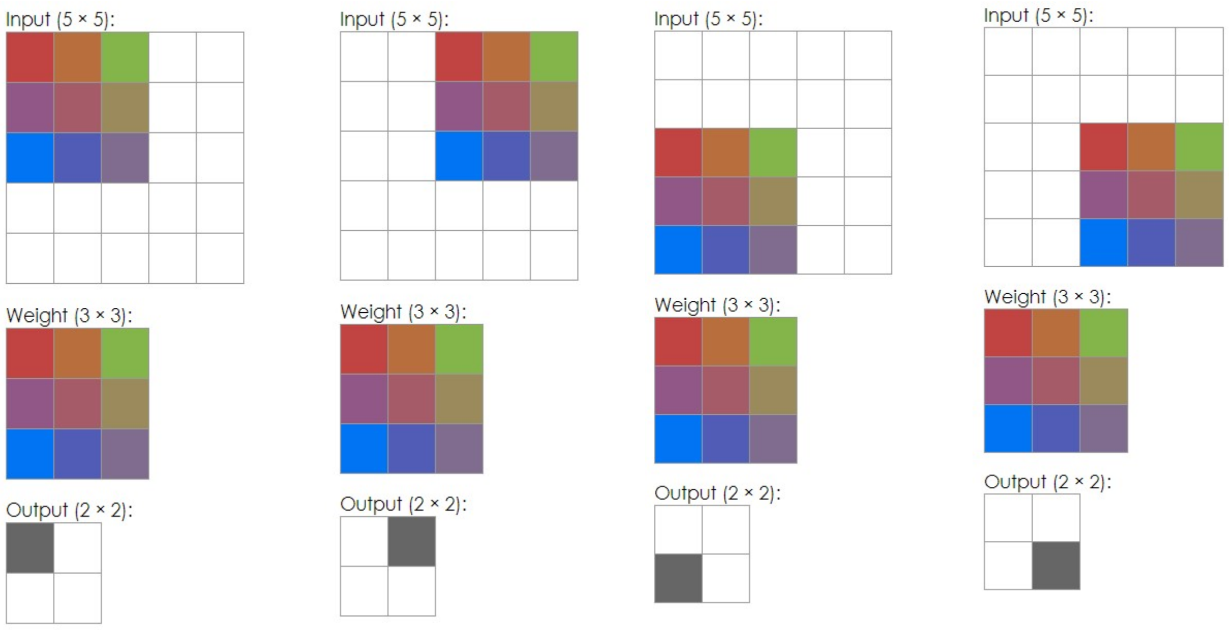
Let us visualize the movement of a kernel of same size with different stride values for an image(5x5) and a kernel of size 3x3 :

Stride : 1



The kernel will move over only one pixel at time, whereas if we set stride = 2 :

Stride : 2



⟶ The grid size - (height x width) defines the number of pixels covered at once, and different grid sizes will impact the size of the output.

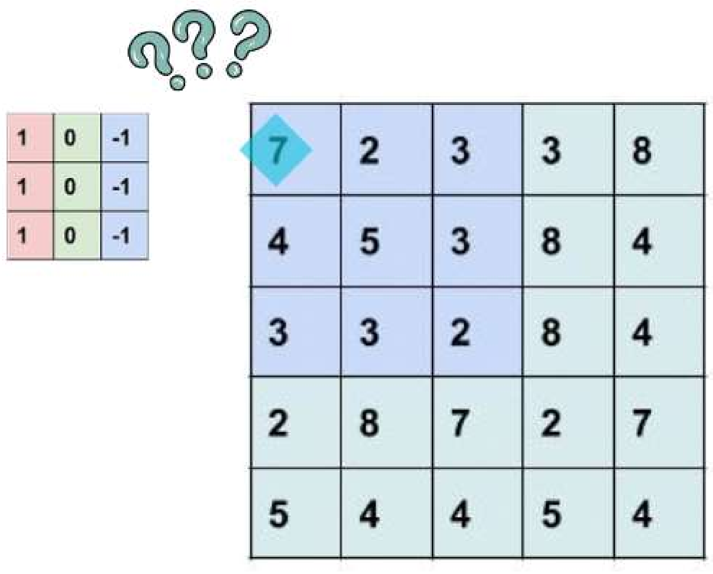
⟶ Typically, the kernel does not need to be square, and the kernel sizes are odd numbers so there is a 'center' pixel.

⟶ The problem at this point is that the kernels will not be able to reach the pixels at the edges and the corners. Since there are no pixels outside of the image around the edge pixels, the kernel is not able to center on the pixels at the edges.

This problem is called the Edge Effect.

The Edge Effect

⟶ As we move a kernel across our image, the problem comes with working on the edge pixels :



⟶ It's possible that we do not put as much weight along each one of the edges.

⟶ Due to this, the edges and corners may not have as much impact on detecting features, while the central pixels are covered more.

As a result :

⟶ Edges might appear blurred or distorted.

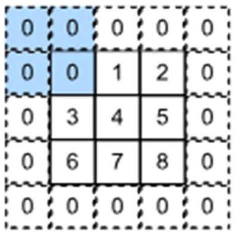
⟶ In some cases, unexpected patterns or values might appear at the edges.

⟶ <u>Information obtained with the edge effect problem will be inconsistent.</u>

Padding

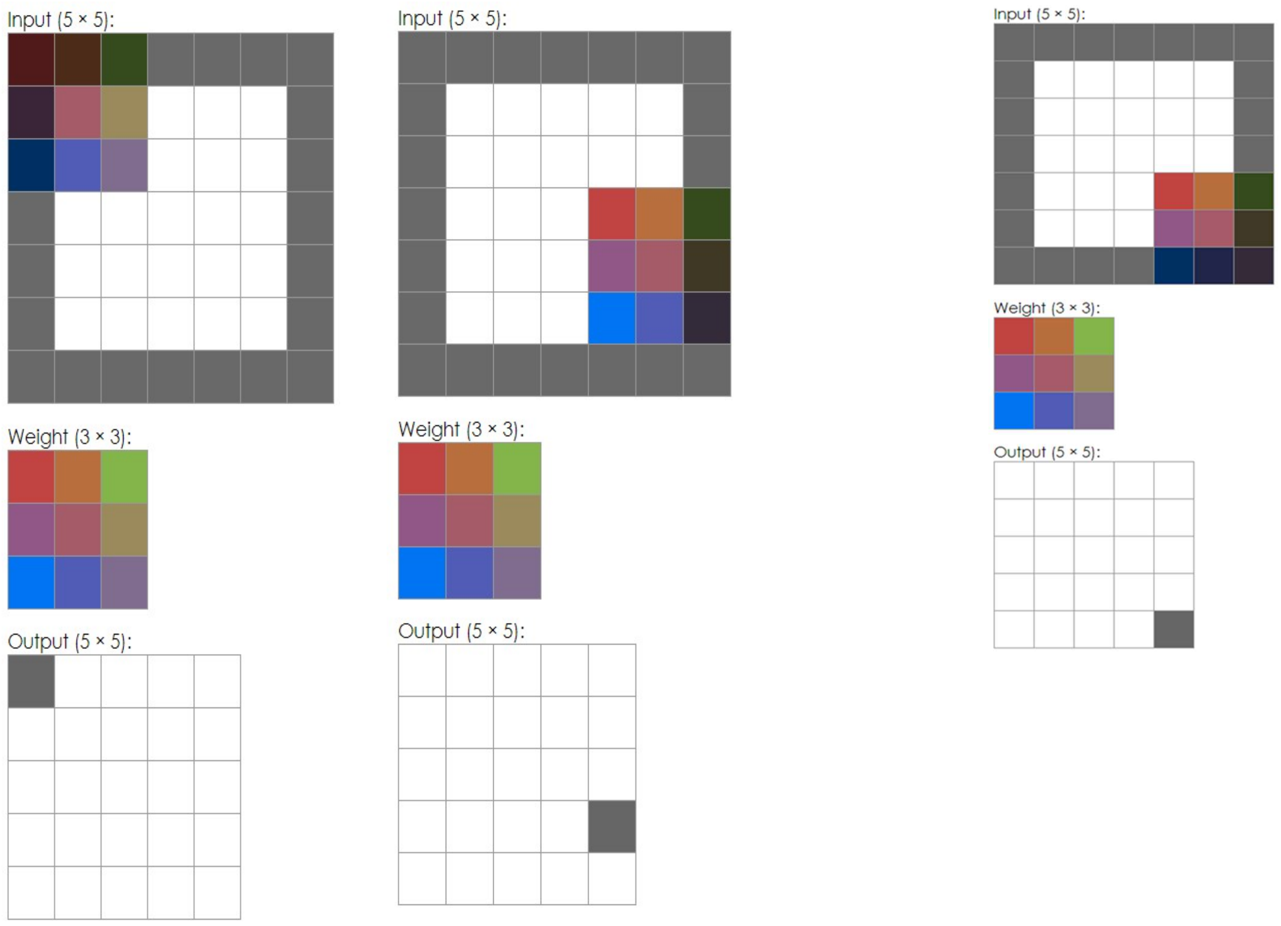
The edge effect can be fixed by padding.

Padding adds extra pixels around the frame, so pixels from the original image become center pixels as the kernel moves across the image.



⟶ Added pixels are typically of value zero, so they do not involve in the convolution operation, but allow the edge pixels to be convoluted.

⟶ Let us visualize the effect of padding :



By padding with 0-value pixels, the edge pixels from the original image become center pixels for the kernel and can be properly convolved.

Thank You !

If you found this article helpful, please do leave a like and comment any feedback you feel I could use.

While I’m working on the next project, I’ll be over at :

www.linkedin.com/in/shivang-kainthola-2835151b9/

shivang.kainthola64@gmail.com

https://shivangkainthola28.medium.com/

https://github.com/HeadHunter28