1. **What is NN and why is it not preferred for image classification**

In regular neural network, the input is transformed through a series of hidden layers having multiple neurons. Each neuron is connected to all the neurons in the previous and the following layers. This arrangement is called a fully connected layer and the last layer is the output layer

where the input is an image, we use convolutional neural network because the regular fully connected neural networks don’t work well. This is because if each pixel of the image is an input then as we add more layers the number of parameters increases exponentially.

**Consider an example** where we are using a **three-colour** channel image with size 1 megapixel **(1000 height X 1000 width)** then our input will have **1000 X 1000 X 3** totalling to 3000000 number of Parameters.

So, the regular neural network is not scalable for image classification as processing such a large input is computationally very expensive and not feasible.

1. **What is Convolution**

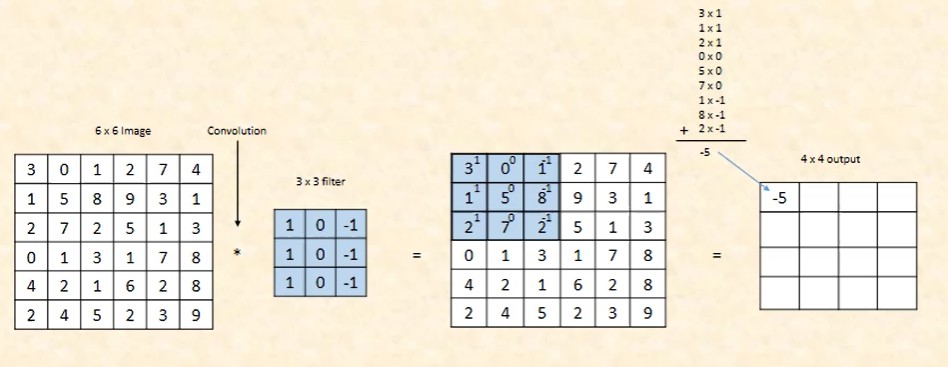
**Convolution** is a mathematical operation on two functions to produce a third function that expresses how the shape of one is modified by the other. In a neural network, we will perform the convolution operation on the input image matrix to reduce its shape.

In the below example, we are convolving a 6 x 6 grayscale image with a 3 x 3 matrix called filter or kernel to produce a 4 x 4 matrix

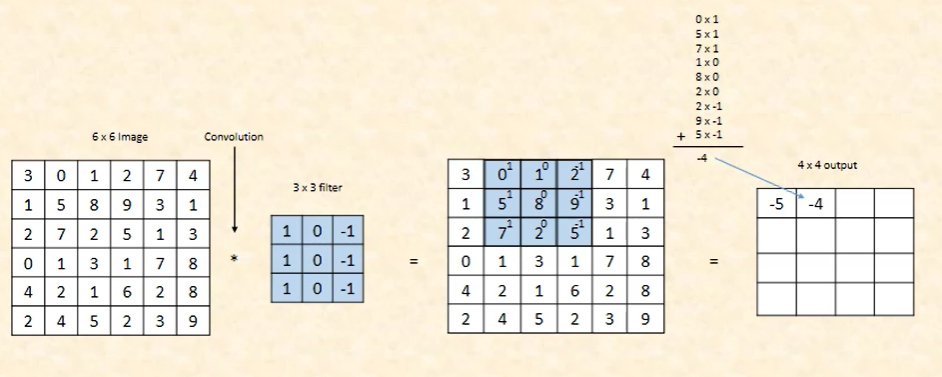
**Steps:**

1. we will take the dot product between the filter and the first 9 elements of the image matrix and fill the output matrix
2. Then we will slide the filter by one square over the image from left to right, from top to bottom and perform the same calculation.

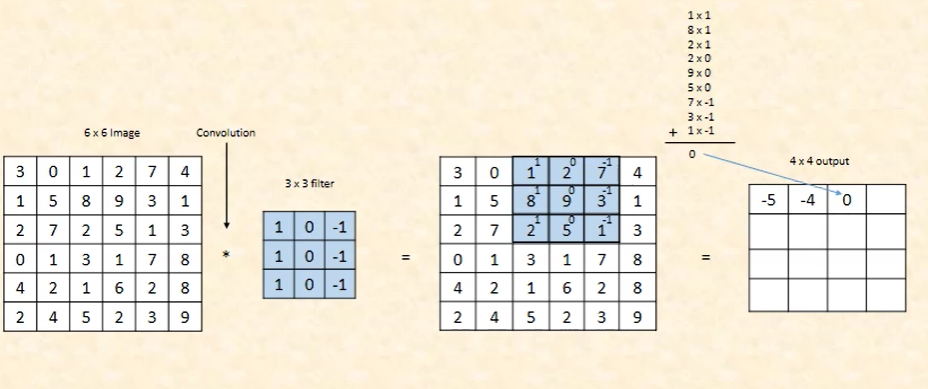
**1st Pass**



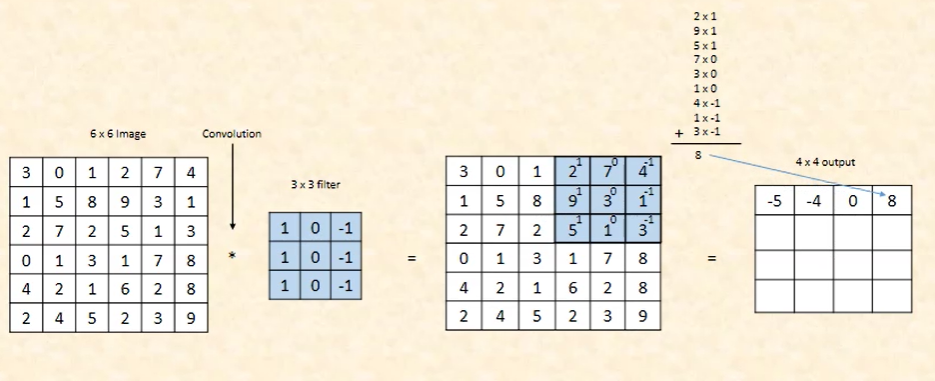
**2nd Pass**



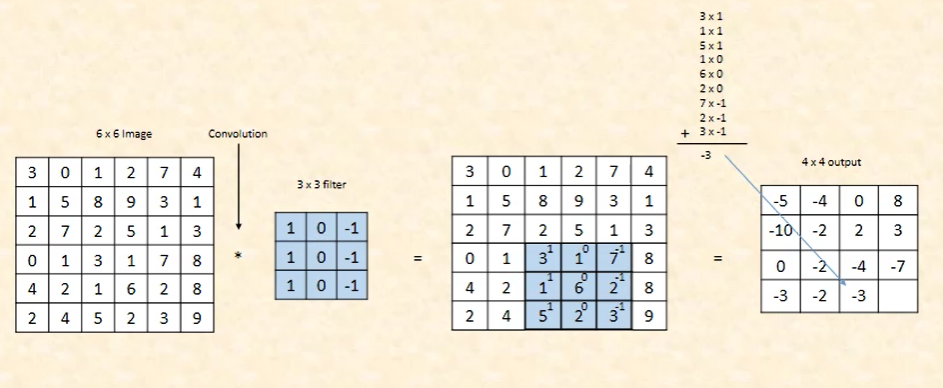
**3rd Pass**



**4th Pass**



**Similarly, for the last Pass**



1. **Challenges**
2. **Shrinking output**

One of the big challenges with convolving is that our image will continuously shrink if we perform convolutional operations in multiple layers.

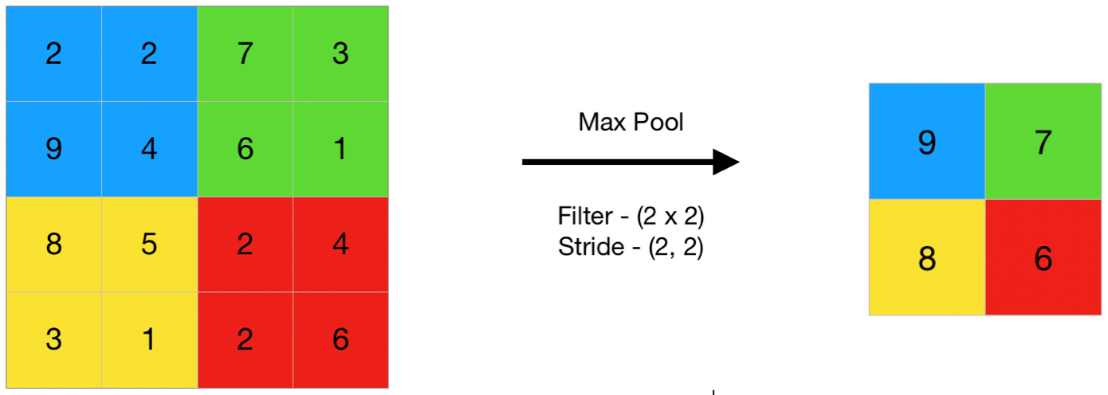
Let’s say if we have 100 hidden layers in our deep neural network and we perform convolution operation in every layer than our image size will shrink a little bit after each convolutional layer.

1. **Pooling**

A pooling layer is another building block of a CNN. Pooling. Its function is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network

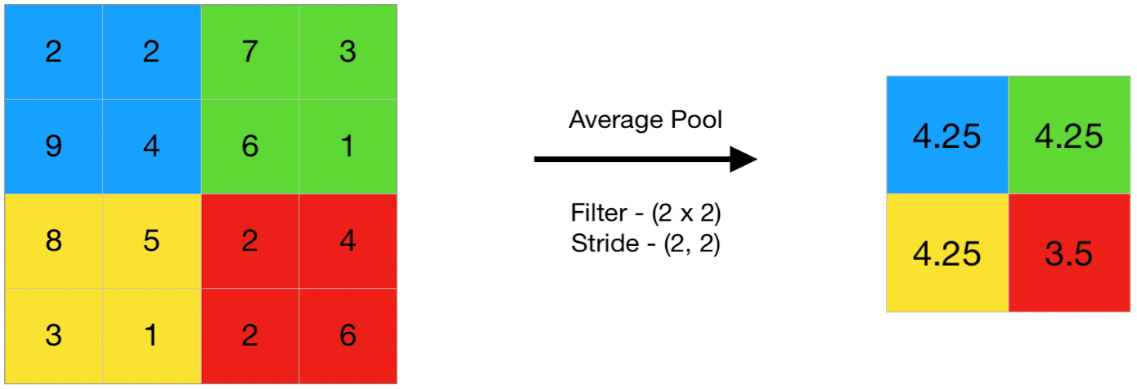
* **Max Pooling**

We will take the maximum value from each block and capture it in our new matrix.



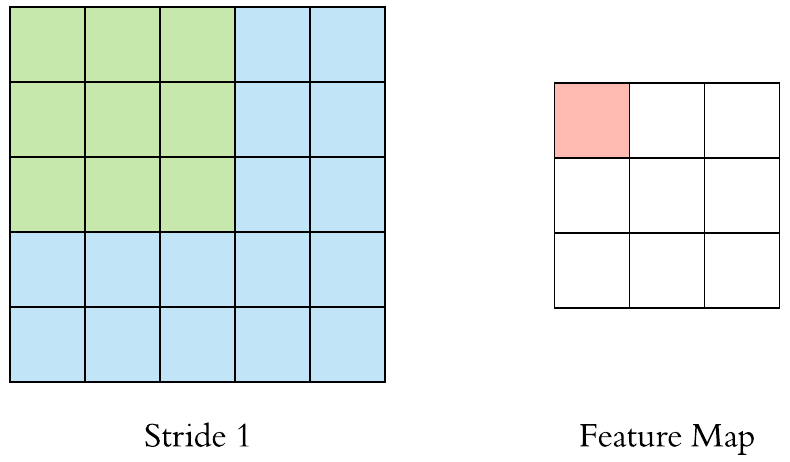
* **Average Pooling**

we take the average of each of the blocks instead of the maximum value.



1. **Stride**

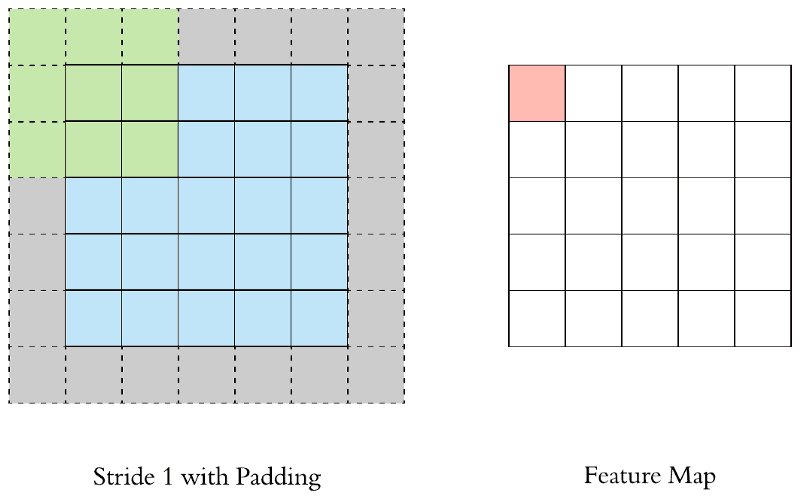
Stride denotes how many steps we are moving in each steps in convolution. By default it is one

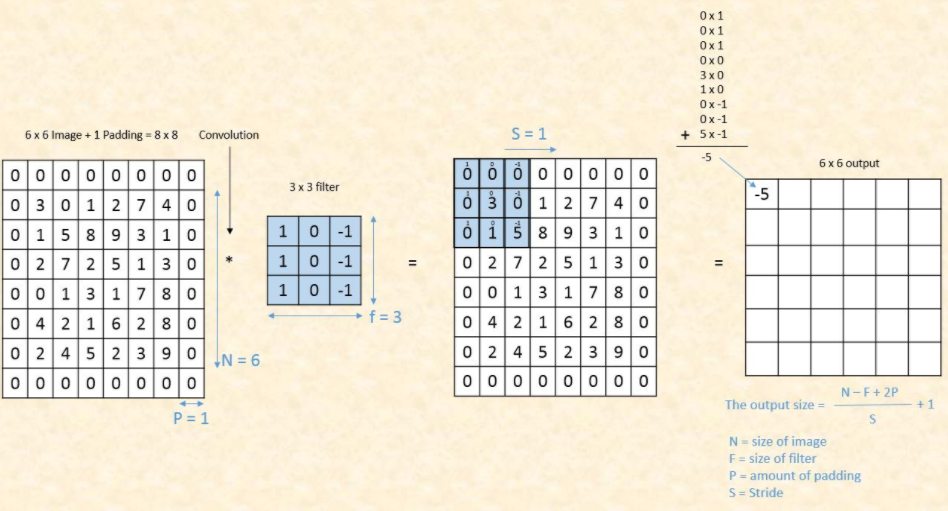
.

1. **Padding**

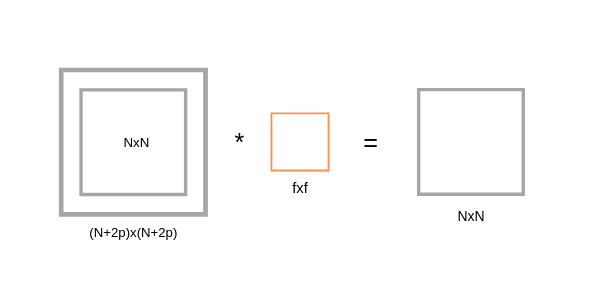
In order to solve the problems of shrinking output and data lost from the image corners, we pad the image with additional borders of zeros called zero padding. The size of the zero padding is a hyperparameter. This allows us to control the spatial size of the output image.

Let P = 1 as the amount of padding that we require, then the image output size is given by the following.





We can see that by using zero padding as 1, we have preserved the size of the original image i.e (6\*6) image.



**Convolution Over RGB Images**

f our image is RGB then the dimensions will be 6 X 6 X 3 where 3 denotes the number of color channels. To detect the features in RGB images we use filters with 3 dimensions.

