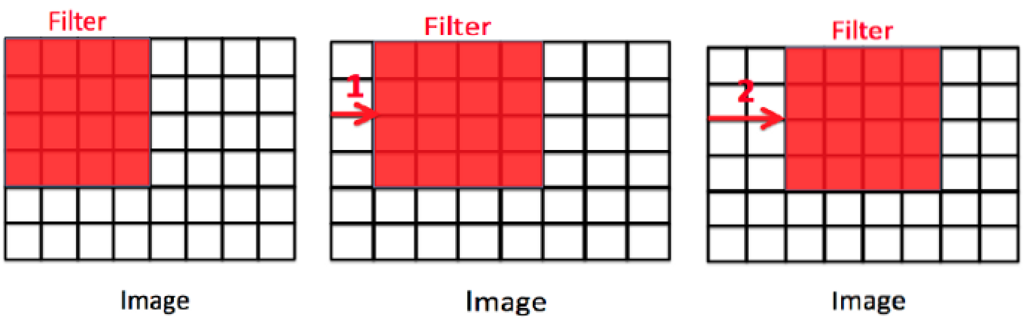
LAB-07

21BDA47

Ques-2) What is Stride, Padding & Pooling? Explain with an example.

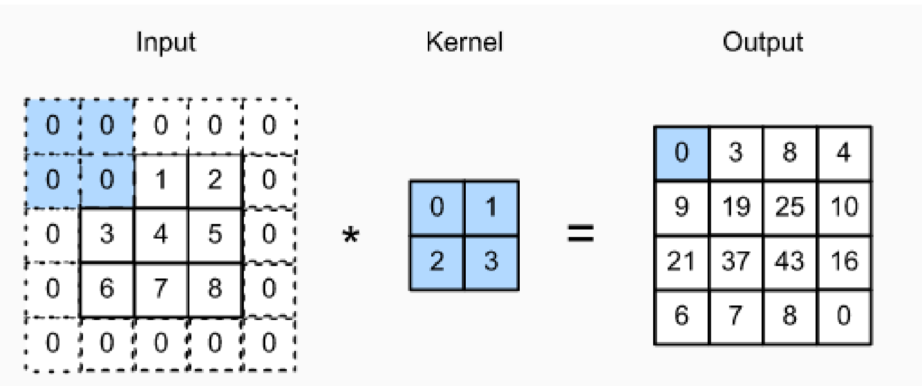
Strides

The pixels are sent to the input matrix when the array is constructed. The strides are the number of pixels that turn to the input matrix. We shift the filters to 1 pixel at a time when the number of strides is 1. The filters are carried to the next two pixels when the number of strides is 2, and so on. They play a crucial role in regulating the features that might be overlooked during flattening the image since they regulate the convolution of the filter against the input. They represent how many steps we take during each convolution. The convolution is depicted in the following figure as it would operate.



Padding

CNN is made in large part by the padding. The image is smaller than it was originally after the convolution operation. We don't want our original image to be reduced in size at each stage of the image classification task's numerous convolution layers. Second, there is an overlap because the kernel crosses the original picture through the central layer more often than the outer layers.



Pooling

Another CNN building piece, the pooling layer, is essential to the pre-processing of images. In the pre-process, if the image is too huge, the size is reduced by lowering the number of parameters. The previous layers are used to create the downscaled image as the picture is downsized, which also reduces the pixel density. Basically, its purpose is to gradually shrink the image's spatial size in order to lessen network complexity and computing expense.

Reference link -
https://www.codingninjas.com/codestudio/library/convolution-layer-padding-stride-and-pooling-in-cnn

Ques-4) What is Stride, Padding & Pooling? Explain with an example.

Machine learning algorithms need a sample dataset to train the model when they are created. The model, however, may begin to learn the "noise"—or irrelevant information—within the dataset if it trains on sample data for an excessively long time or if the model is overly complex. The model becomes "overfitted" and unable to generalise successfully to new data when it memorises the noise and fits the training set too closely. A model won't be able to carry out the classification or prediction tasks that it was designed for if it can't generalise successfully to new data. High variance and low error rates are reliable signs of overfitting.

How to avoid overfitting

- **Train with more data:** By giving additional possibilities to isolate the dominant link between the input and output variables, expanding the training set to include more data might improve the model's accuracy.
- **Data augmentation:** Although it is preferable to add clean, pertinent data to your training set, noisy data may occasionally be incorporated in order to increase the stability of a model. However, this approach needs to be used carefully.
- **Feature selection:** Many of the parameters or features that you use to develop a model to predict a specific outcome may often be redundant with other features. The act of selecting the most crucial features from the training data and removing the redundant or unnecessary ones is known as feature selection.
- **Regularization:** The bigger coefficient input parameters receive a "penalty" through regularisation, which subsequently reduces the variance in the model. While there are several regularisation techniques, including dropout, Lasso regularisation, and L1 regularisation, they all aim to find and remove the noise in the data.
- **Ensemble methods:** A number of classifiers, such as decision trees, are used in ensemble learning techniques, and their predictions are combined to determine the most common outcome. The most popular ensemble techniques are boosting and bagging.