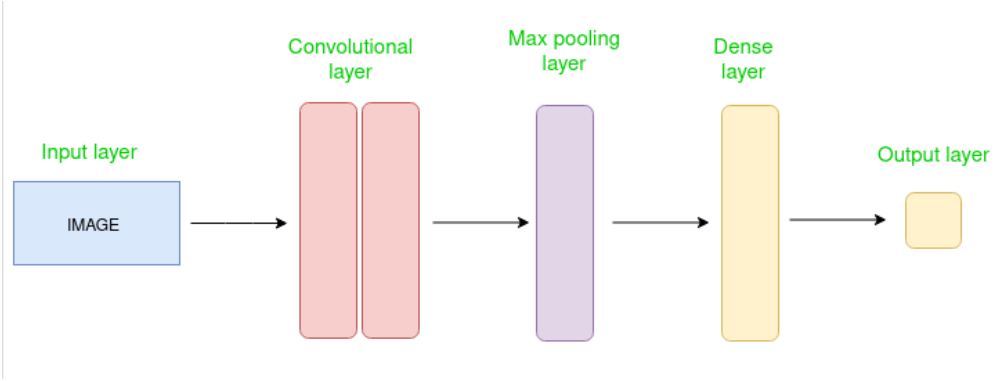
Convolution Neural Networks CNN

A **Convolutional Neural Network (CNN)** is a type of **Deep Learning neural network** architecture commonly used in Computer Vision.

Convolutional Neural Network (CNN) is the extended version of artificial neural networks (ANN) which is predominantly used to extract the feature from the grid-like matrix dataset. For example visual datasets like images or videos where data patterns play an extensive role.

Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.



The Convolutional layer applies filters to the input image to extract features, the Pooling layer downsamples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

Layers Used to Build ConvNets

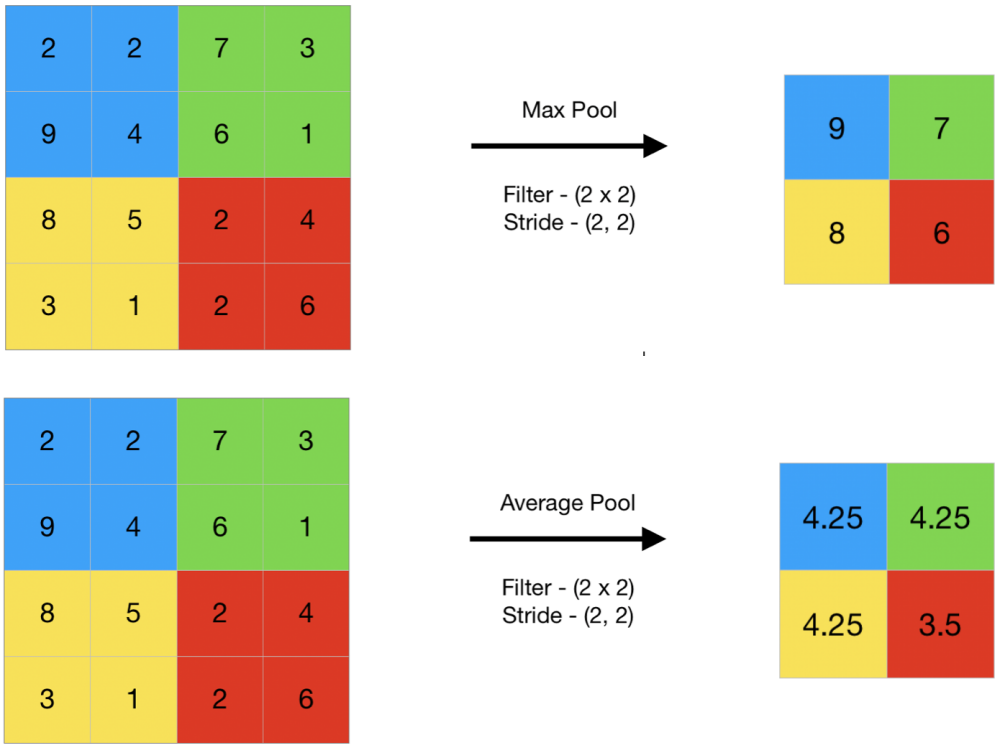
A complete Convolution Neural Networks architecture is also known as covnets. A covnets is a sequence of layers, and every layer transforms one volume to another through a differentiable function.

Types of layers: datasets

Let's take an example by running a covnets on of image of dimension 32 × 32 × 3.

- **Input Layers:** It's the layer in which we give input to our model. In CNN, Generally, the input will be an image or a sequence of images. This layer holds the raw input of the image with width 32, height 32, and depth 3.
- **Convolutional Layers:** This is the layer, which is used to extract the feature from the input dataset. It applies a set of learnable filters known as the kernels to the input images. The filters/kernels are smaller matrices usually 2×2, 3×3, or 5×5 shape. it slides over the input image data and computes the dot product between kernel weight and the corresponding input image patch. The output of this layer is referred as feature maps. Suppose we use a total of 12 filters for this layer we'll get an output volume of dimension 32 × 32 × 12.

- **Activation Layer:** By adding an activation function to the output of the preceding layer, activation layers add nonlinearity to the network. It will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are **RELU**: max(0, x), **Tanh**, etc. The volume remains unchanged hence output volume will have dimensions 32 × 32 × 12.
- **Pooling layer:** This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Common types of pooling layers are **max pooling** , and **average pooling**. If we use a max pool with 2 × 2 filters and stride 2, the resultant volume will be of dimension 16×16×12.



- **Flattening:** The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layer for categorization or regression.
- **Fully Connected Layers:** It takes the input from the previous layer and computes the final classification or regression task.
- **Output Layer:** The output from the fully connected layers is then fed into a logistic function for classification tasks like **sigmoid** or **softmax** which converts the output of each class into the probability score of each class.

Student Name **Roll No.** **Section:**

Q1. Input image is given, you are required to pass it through forward pass of CNN model with 1 conv layer consists of 2 filters, stride = 1, padding = 1 and then apply mean pooling layer with filter size 2 and stride = 2, apply flattening then and pass it through an mlp with one hidden layer of size 6 and and output layer of size 3. Activation function for hidden layer will be tanh and for output layer softmax. Weight matrices are given for conv layer. For all the weights and bias between flattened layer and hidden layer use value 0.3 and between hidden layer and output layer use value 0.4.

F1					b1
3	2	1	-1	1	0.5
0	5	2	3	0	
-1	1	-2	-3	-1	
5	2	0	1	0	
4	2	1	0	-2	
F2					b2
1	2	3	4	5	0.8
0	9	8	7	6	
3	2	1	0	4	
9	8	7	6	5	
-1	0	-2	4	8	

58	36	5	30	37	62	7	81
42	31	9	11	14	51	62	52
23	90	71	29	10	71	49	55
28	82	30	7	70	55	52	19
52	47	90	29	53	28	96	8
85	45	95	95	9	32	54	48
49	71	26	75	41	50	17	15
63	29	49	77	18	28	40	51

$$S(y)_i = \frac{\exp(y_i)}{\sum_{j=1}^{n} \exp(y_j)} \qquad tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

CNN Example.pdf