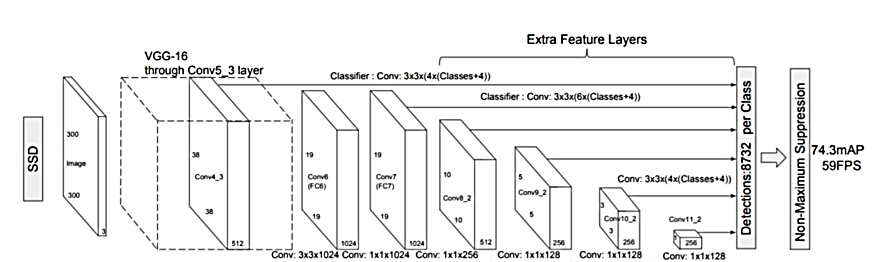
**What is Object Detection?**

Object Detection in Computer Vision is detecting and predicting objects and localizing their area in a given frame.

Object Detection is based on image classification. The top Object Detection Networks are:

1. Faster R-CNN
2. YOLO
3. SSD

**Single Shot MultiBox Detector (SSD)**

SSD Object Detection follows this 2 steps:

1. extracts feature map using a base deep learning network, which are CNN based classifiers
2. applies convolution filters to finally detect objects

Our implementation uses MobileNet as the base network.

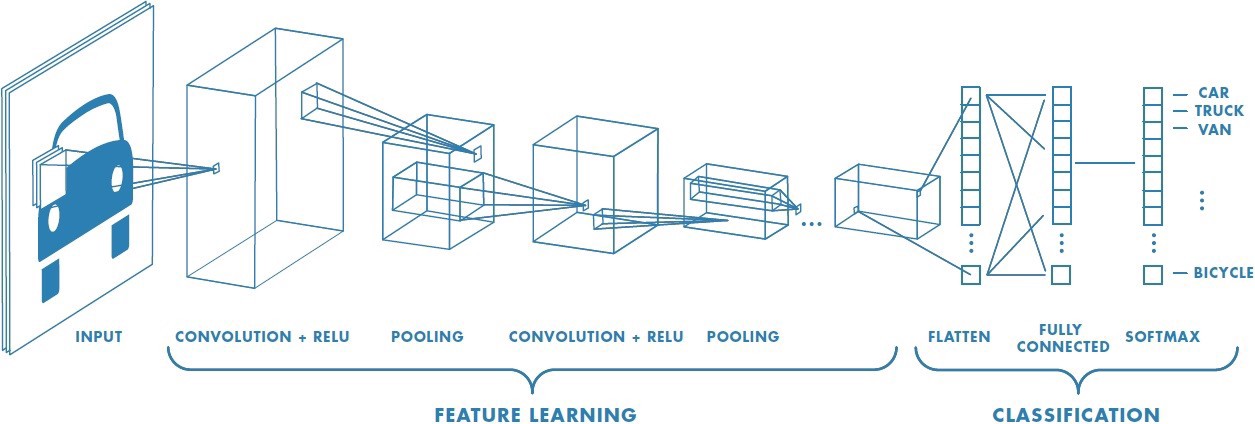
SSD uses VGG16 to extract feature maps. Then it detects objects using the Conv5\_3 layer.

Then a Series of CNN comes to action.

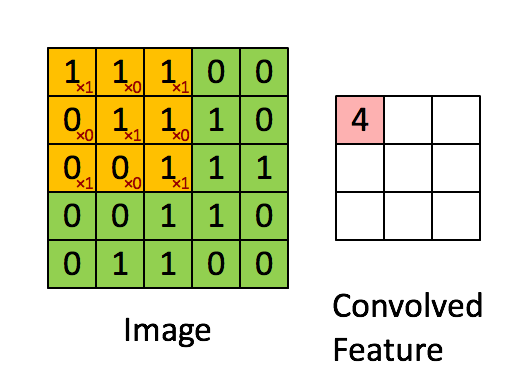
A great way to use deep learning to classify images is to build a convolutional neural network (CNN).

Computers see images using pixels. Pixels in images are usually related. For example, a certain group of pixels may signify an edge in an image or some other pattern (Letter or number). Convolutions use this to help identify images.

**So How a Convolutional Neural Networks work?**



We take the example below:

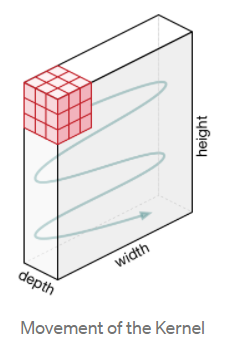


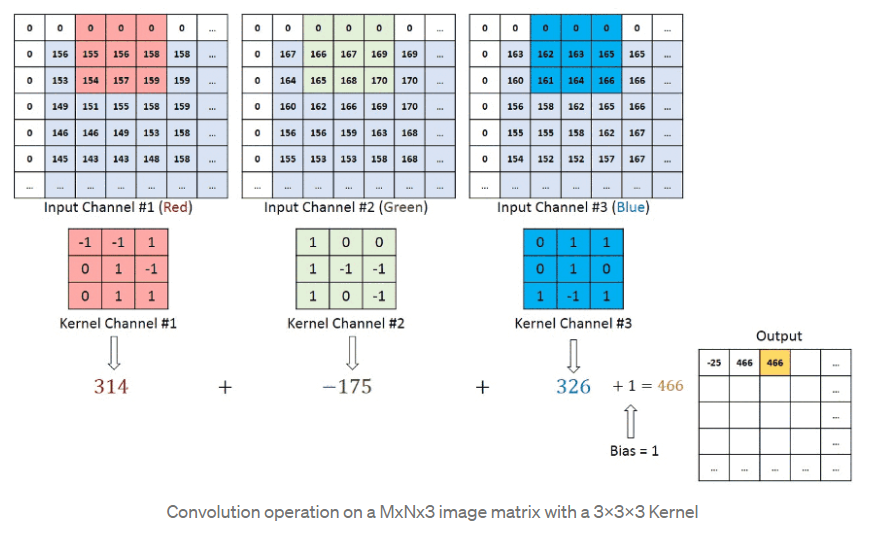
In the above demonstration, the green section resembles our 5x5x1 input image, I. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter, K, represented in the color yellow. We have selected K as a 3x3x1 matrix.

Kernel/Filter,

K = 1 0 1  
 0 1 0  
 1 0 1

The Kernel shifts 9 times because of Stride Length = 1 (Non-Strided), every time performing a matrix multiplication operation between K and the portion P of the image over which the kernel is hovering.





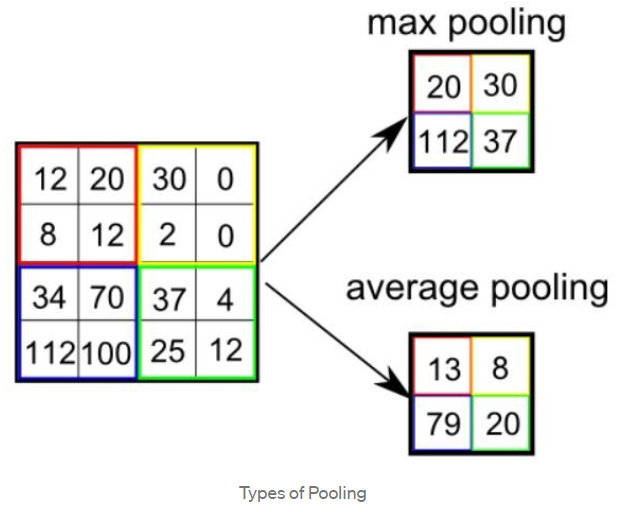
The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. CNN need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the dataset.

**Pooling Layer**

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

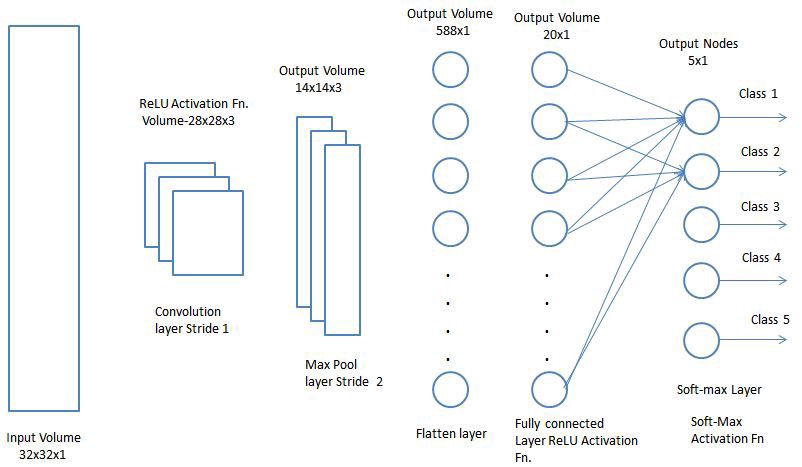


The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power.

After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

**Classification — Fully Connected Layer (FC Layer)**

Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a **feed-forward neural network** and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the **Softmax** Classification technique.



**What is Mobilenet-ssd & Caffe?**

The Mobilenet-ssd model is a Single-Shot multibox Detection (SSD) network intended to perform object detection. This model is implemented using the Caffe framework.

Caffe is a deep learning framework developed by Berkeley AI Research and community contributors. Caffe was developed as a faster and far more efficient alternative to other frameworks to perform object detection. Caffe can process 60 million images per day with a single NVIDIA K-40 GPU. That is 1 ms/image for inference and 4 ms/image for learning.

*NB:* This Single Shot Detector (SSD) object detection model uses Mobilenet as backbone and can achieve fast object detection optimized for mobile devices.