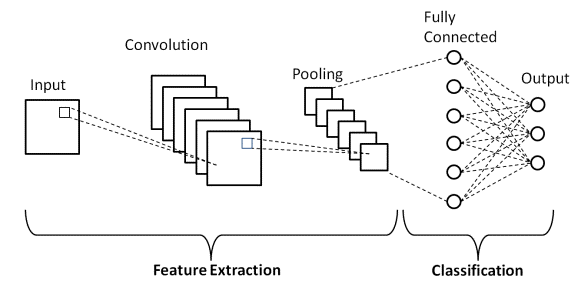
Comparison of CNN and Yolo

1. Architecture: CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers extract features from input images, and the fully connected layers classify these features. CNNs typically use sliding convolutional filters to detect local patterns in the input data. YOLO is a single-shot object detection system that processes the entire image in a single pass. It divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell.
2. Speed: YOLO performs object detection in a single pass, making it significantly faster than traditional CNN-based approaches. It sacrifices some accuracy for speed, making it suitable for real-time applications where low latency is crucial. Traditional CNN-based object detection methods, such as Faster R-CNN or SSD (Single Shot MultiBox Detector), require multiple passes over the image and are computationally expensive. They have relatively slower inference times.
3. Accuracy: YOLO sacrifices some accuracy for speed. While it may not achieve the same level of accuracy as some state-of-the-art CNN-based detectors, it still performs well in real-world scenarios and provides a good trade-off between speed and accuracy. CNN-based object detectors achieve high accuracy due to their multi-stage architecture. By leveraging region proposal networks and anchor boxes, they can precisely localize objects and classify them accurately.
4. Object Detection: CNN-based detectors, with their multi-stage approach, are well-suited for detecting small objects and objects with complex shapes. They excel in scenarios where precise localization and detailed object recognition are important. YOLO is designed for real-time object detection and performs better when detecting larger objects. It may struggle with accurately localizing small objects and objects with intricate details.
5. Training: Training CNN-based object detectors often involves multiple stages, including region proposal generation, feature extraction, and classification. It can be more complex and time-consuming compared to YOLO. YOLO is relatively simpler to train compared to traditional CNN-based detectors. It can be trained end-to-end, requiring fewer steps and computations.

**Layers in CNN**

****

### 1. Convolutional Layer

### This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM). The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

2. Pooling Layer

The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. In Max Pooling, the largest element is taken from the feature map. Average Pooling calculates the average of the elements in a predefined size Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer. This CNN model generalizes the features extracted by the convolution layer, and helps the networks to recognise the features independently. With the help of this, the computations are also reduced in a network.

### 

### 

### 

### **3. Fully Connected Layer**

### The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place.

### **4. Dropout**

### Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model’s performance when used on new data. To overcome this problem, a dropout layer is utilized wherein a few neurons are dropped from the neural network during the training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network. Dropout results in improving the performance of a machine learning model as it prevents overfitting by making the network simpler. It drops neurons from the neural networks during training.

### **5. Activation Functions**

### They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. It decides which information of the model should fire in the forward direction and which ones should not at the end of the network. It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax, tanH and the Sigmoid functions. For a binary classification CNN model, sigmoid and softmax functions are preferred and for a multi-class classification, generally softmax is used. Activation functions in a CNN model determine whether a neuron should be activated or not. It decides whether the input to the work is important or not to predict using mathematical operations.