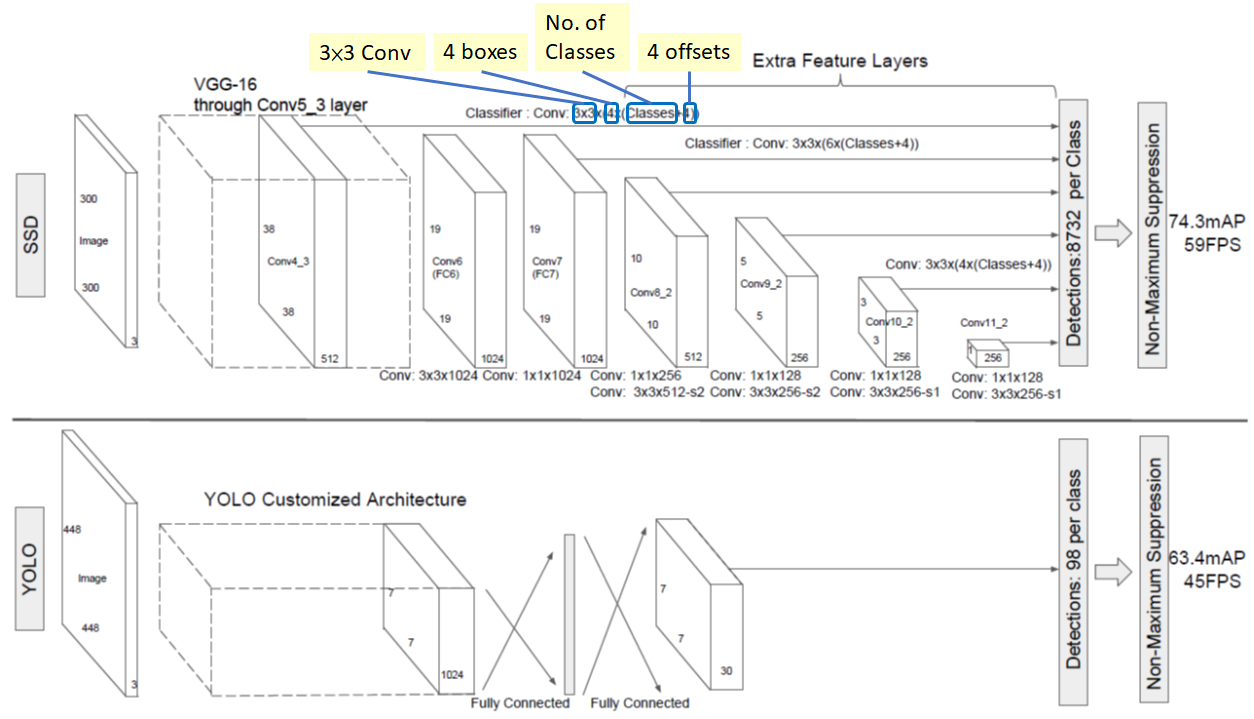
SSD MODEL ARCHITECTURE

Single shot detectors use multiple layers that provide a finer accuracy on objects with different scales. (Each deeper layer will see bigger objects).



**SSD framework**:

* SSD only needs an input image and ground truth boxes for each object during training. In a convolution manner. We evaluate a small of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8 × 8 and 4 × 4).
* For each default box, we predict both the shape offsets and the confidences for all object categories ((c1, c2, · · · , cp)).
* At training time, we first match these default boxes to the ground truth boxes. For example, we have matched two default boxes with the cat and one with the dog, which are treated as positives and the rest as negatives.
* The model loss is a weighted sum between localization loss (e.g. Smooth L1 [6]) and confidence loss (e.g. Softmax)
* The SSD normally start with a VGG on Resnet pre-trained model that is converted to a fully convolution neural network.
* Then we attach some extra conv. layers, which will actually help to handle bigger objects. The SSD architecture can in principle be used with any deep network base model.
* One important point to notice is that after the image is passed on the VGG network, some conv layers are added producing feature maps of sizes 19x19, 10x10, 5x5, 3x3, 1x1. These, together with the 38x38 feature map produced by VGG’s conv4\_3, are the feature maps which will be used to predict bounding boxes.
* There the conv4\_3 is responsible to detect the smallest objects while the conv11\_2 is responsible for the biggest objects*.*
* Some activations are taken from the network and passed to a specialized sub-network that should work as a classifier and localizer.
* During prediction we use a Non-maxima suppression algorithm to filter the multiple boxes per object that may appear.

**SSD Model Approach**

* The SSD approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections.
* The early network layers are based on a standard architecture used for high quality image classification (truncated before any classification layers), which we will call the base network1 . We then add auxiliary structure to the network to produce detections with the following key features: Multi-scale feature maps for detection We add convolutional feature layers to the end of the truncated base network. These layers decrease in size progressively and allow predictions of detections at multiple scales.
* The convolutional model for predicting detections is different for each feature that operate on a single scale feature map. Each added feature layer (or optionally an existing feature layer from the base network) can produce a fixed set of detection predictions using a set of convolutional filters.
* For a feature layer of size m × n with p channels, the basic element for predicting parameters of a potential detection is a 3 × 3 × p small kernel that produces either a score for a category, or a shape offset relative to the default box coordinates.
* Matching strategy At training time we need to establish the correspondence between the ground truth and the default boxes. For each ground truth box we are selecting from default boxes that vary over location, aspect ratio, and scale. We begin by matching each ground truth box to the default box with the best jaccard overlap. This is the matching approach used by the original MultiBox and it ensures that each ground truth box has exactly one matched default box. Unlike MultiBox, we then match default boxes to any ground truth with jaccard overlap higher than a threshold (0.5).
* Adding these matches simplifies the learning problem: it allows the network to predict high confidences for multiple overlapping default boxes rather than requiring it to pick only the one with maximum overlap.
* Anchors are a collection of boxes overlaid on the image at different spatial locations, scales and aspect ratios that act as reference points on the ground truth images. It's like the Yolo idea where each cell on the activation map has multiple boxes.
* **Non Maximum Support**: Sort all boxes of a class using confidence scores. Calculate Jaccard Index of first box with every other box. If overlap > 0.45, remove the other box. Otherwise keep the other box. Repeat the above process for each box in sorted order.
* Basically we need to compare if the Intersect Over Union (ioU) between the prediction and the ground truth is bigger than some threshold (ex > 0.5)

