### **Object Detection using SSD**

SSD is designed for object detection in real-time. Faster R-CNN uses a region proposal network to create boundary boxes and utilizes those boxes to classify objects. While it is considered the state-of-the-art in accuracy, the whole process runs at 7 frames per second. Far below what a real-time processing needs. SSD speeds up the process by eliminating the need of the region proposal network. To recover the drop in accuracy, SSD applies a few improvements including multi-scale features and default boxes. These improvements allow SSD to match the Faster R-CNN’s accuracy using lower resolution images, which further pushes the speed higher.

## **Single Shot MultiBox Detector for real-time processing**

The SSD object detection composes of 2 parts

-->Extract feature maps, and

-->Apply convolution filters to detect objects.

### **Extract feature maps**

SSD uses VGG16 to extract feature maps. Then it detects objects using the Conv4\_3 layer. For illustration, we draw the Conv4\_3 to be 8 × 8 spatially (it should be 38 × 38). For each cell (also called location), it makes 4 object predictions.

### **Convolutional predictors for object detection**

SSD does not use a delegated region proposal network. Instead, it resolves to a very simple method. It computes both the location and class scores using small convolution filters. After extracting the feature maps, SSD applies 3 × 3 convolution filters for each cell to make predictions. (These filters compute the results just like the regular CNN filters.) Each filter outputs 25 channels: 21 scores for each class plus one boundary box.

**Steps of Detector:**

**Imports**

-->Importing all the requirement libraries like Numpy ,OpenCV, Tensorflow ,matplotlib....

# **Model preparation:**

-->Any model exported using the export\_inference\_graph.py tool can be loaded here simply by changing PATH\_TO\_FROZEN\_GRAPH to point to a new .pb file.

-->we use an “SSD” with Mobilenet model here.

-->Download Model

-->Load a (frozen) Tensorflow model into memory.

## **Loading label map:**

-->Label maps map indices to category names, so that when our convolution network predicts 5, we know that this corresponds to airplane. Here we use internal utility functions, but anything that returns a dictionary mapping integers to appropriate string labels would be fine.

--> Run inference through the network and gather predictions from output layers.

-->For each detetion from each output layer get the confidence, class id, bounding box params and ignore weak detections.

-->Draw bounding box and display output image

# **Conclusion:**

SSD is a single-shot detector. It has no delegated region proposal network and predicts the boundary boxes and the classes directly from feature maps in one single pass.

To improve accuracy, SSD introduces:

* small convolutional filters to predict object classes and offsets to default boundary boxes.
* separate filters for default boxes to handle the difference in aspect ratios.
* multi-scale feature maps for object detection.

SSD can be trained end-to-end for better accuracy. SSD makes more predictions and has a better coverage on location, scale and aspect ratios. With the improvements above, SSD can lower the input image resolution to 300 × 300 with a comparative accuracy performance. By removing the delegated region proposal and using lower resolution images, the model can run at real-time speed and still beats the accuracy of the state-of-the-art Faster R-CNN.