Single Shot Multi Box Detectors (SSD)

Friday, December 7, 2018 4:39 PM

What is SSD?
It is a machine learning algorithm used for the computer vision applications.

What is the purpose?
This algorithm is used to draw bounding boxes around the different objects of interest in an image with vector of values representing probabilities of different class of objects of interest in any particular box.

What is the advantage of SSD?
It is one of the real-time object detection algorithms. It can be run on embedded systems with decent computation capabilities in real time

What is the common pipeline of previous approaches?
Previous approaches involve the pipeline of hypothesizing the bounding boxes, resample pixels or features for each box, and apply the high quality classifier. (e.g. Faster R-CNN)

What is the drawback of the previous approaches that led to the approaches like SSD?
The previous approaches are too computationally intensive for embedded systems, even using high-end hardware, too slow for real-time applications. (e.g. Faster R-CNN operates only at 7 frames per second (FPS))

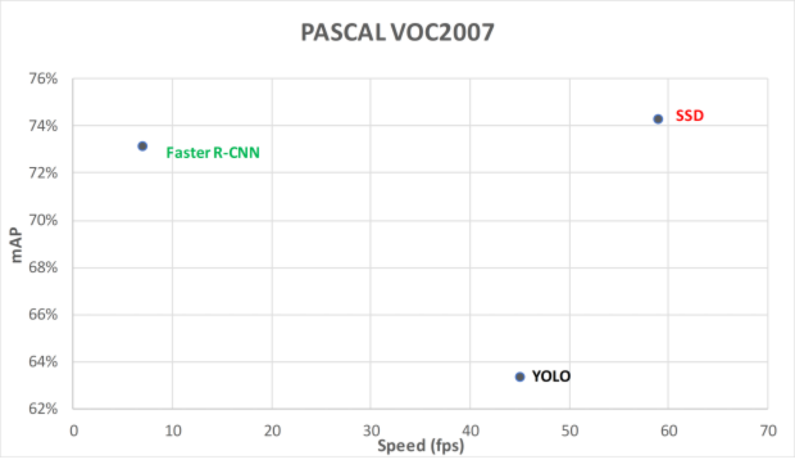
How SSD increases the prediction speed?
Improvement in speed comes from elimination of bounding box proposals and pixel or feature resampling. But this approach is also used by YOLO algorithm.

How it is different in the context of neural network architecture from YOLO ?
SSD discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature (from different depth of convolutional neural network) map location.

How SSD improves the prediction accuracy when compared to similar approaches?
SSD improves the prediction accuracy by using following tweaks

1. Uses small convolutional filter to predict object categories and offsets in bounding box locations, using separate predictors / filters for different aspect ratio detections.
2. Applying these filters to multiple feature maps from the later stages of a network in order to perform detections at multiple scales

What is the evidence that shows the performance of SSD is better than YOLO and Faster R-CNN?
All the three neural networks were tested on PASCAL VOC2007 dataset and the mean average precision (mAP) is plotted below. The algorithm which produces the point on the top right region (*High Processing Speed and High Accuracy*) is desirable and SSD lies in that region.



How the model of the SSD looks like?
SSD approach is based on a feed-forward CNN, which produces a collection of fixed-size bounding boxes and also scores for the presence of object class instances in those boxes, passed through a non-maximum suppression step to produce final detections.

What are the important layers of the SSD and what would they provide?

1. Base Layer:
   The early network layers is part of the standard architecture (truncated before the classification / fully connected layers) used for the high quality image classification. This truncated layer is called as the base layer.

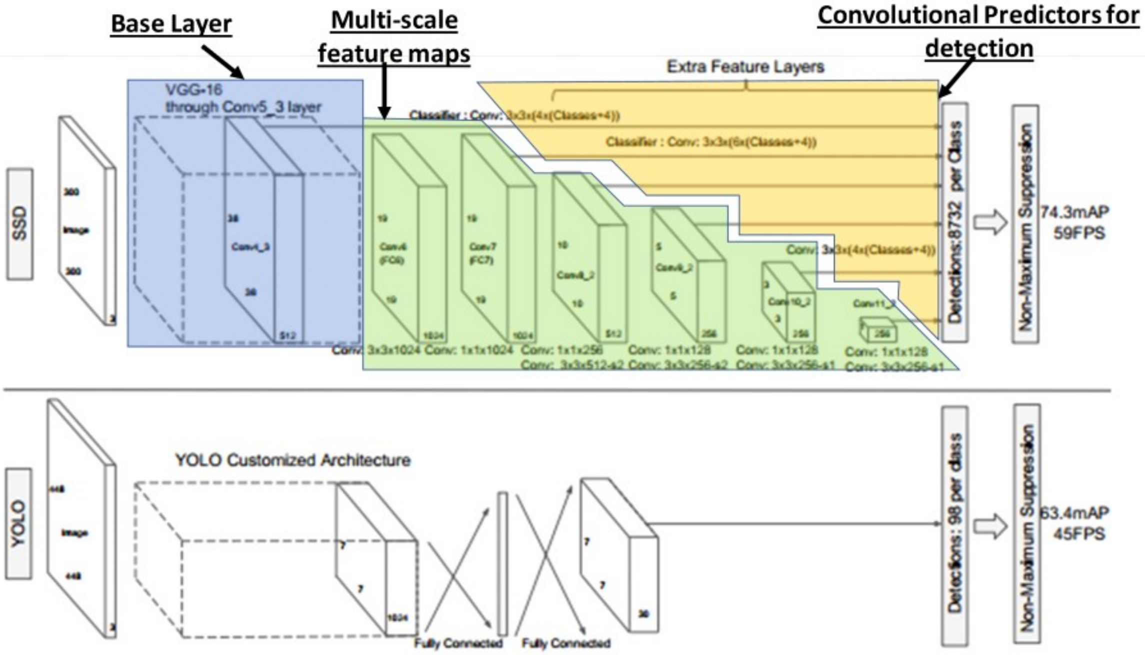
2. Multi-scale feature maps :
   Convolutional feature layers are added to the end of base (truncated) network. These layers decrease in size progressively to allow detections at multiple scales. The convolution model for predicting detections is different for each feature layer. (This approach is not in YOLO, YOLO operates in single scale feature map).

3. Convolutional Predictors for detection:
   Each feature layer of the multi-scale feature maps produce a set of detection predictions using a set of convolutional filters.
   Example: For a feature layer of size m x n with p channels, a small kernel of 3 x 3 x p is applied for each location to produce either a score for a category or a shape offset relative to the default box coordinates.
   Offset values are measured relative to each feature map location.
   (YOLO uses an intermediate fully connected layer instead of a convolutional filter for this step).

4. Default Boxes and aspect ratios:
   Each feature map cell of all feature maps (after the base layer) in the network is associated with a set of default bounding boxes. At each cell the offsets relative to the default box shapes in the cell and the per-class scores (indicate the presence of a class instance) of each default boxes will be predicted. For each box out of 'k' default boxes at a cell, 'c' class scores and the 4

offsets relative to default box shapes have to be computed. So a total of (c+4)k filters have to applied at each location to yield (c+4) x k x m x n outputs for a m x n feature map. This approach of creating different box shapes in several feature maps help to efficiently discretize the shape of possible output box shapes.

Please see the following figures for better understanding



Reference: https://arxiv.org/abs/1512.02325