0.1 SSD: Single Shot MultiBox Detector

- *Algorithm:* SSD: Single Shot MultiBox Detector
- *Input:* A RGB 3 channel picture
- *Complexity:* None
- *Data structure compatibility:* None
- *Common applications:* Artificial intelligence

Problem. SSD: Single Shot MultiBox Detector

Object detection is the task of detecting instances of objects of a certain class within an image. A sample of object detection is shown in Figure 1.

Figure 1: **Detection examples on COCO `test-dev` with SSD512 model.** Each color corresponds to an object category.

SSD [**liu2016ssd**], a single-shot detector for multiple categories, is faster than the previous state-of-the-art for single shot detectors (YOLO [**redmon2016you**]), and more accurate, region proposals and pooling (including Faster R-CNN [**ren2015faster**]). The core of SSD is predicting category scores and box offsets for a fixed set of default bounding boxes using small convolutional filters applied to feature maps. To achieve high detection accuracy it produces predictions of different scales from feature maps of different scales, and explicitly separate predictions by aspect ratio. These design features further improve the speed vs accuracy trade-off by leading to simple end-to-end training and high accuracy, even on low resolution input images.

Description

Model

The SSD approach is based on a feed-forward convolutional network. Network produces a fixed-size collection of bounding boxes and scores for the object instances in those boxes. It then follows by a non-maximum suppression step to produce the final detections. The early network layers called the base network are based on a standard architecture used in image classification. It then add auxiliary structure to the network to produce detections with the following key features:

For multi-scale feature maps for detection, it adds convolutional feature layers to the end of the base network. These layers progressively decrease in size and predicts detections at multiple scales. The convolutional model for predicting detections is different for each feature layer.

For convolutional predictors for detection, each feature layer can use a set of convolutional filters to produce a fixed set of detection predictions . These are on top of the SSD network architecture in Fig. 2. For a $m \times n$ size

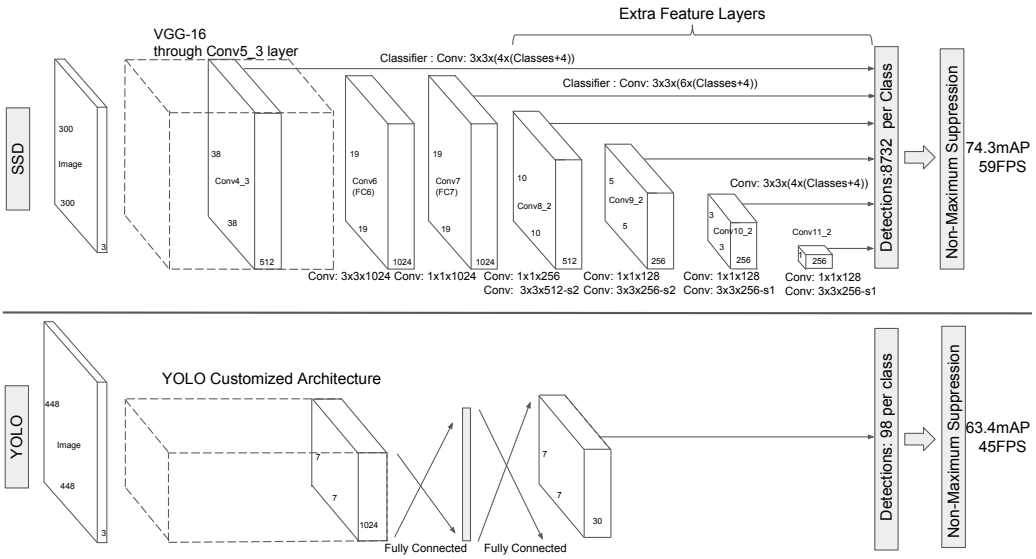

Figure 2: A comparison between two single shot detection models: SSD and YOLO [**redmon2015you**]. Several feature layers are added to the end of a base network in SSD, which predict the offsets to default boxes.

feature layer with p channels, the basic element for predicting parameters of a potential detection is a $3 \times 3 \times p$ small kernel that produces either a score for a category, or a shape offset relative to the default box coordinates. It produces an output value at each of the $m \times n$ locations. The bounding box offset will be then measured.

Training

The SSD training objective is inherited from the MultiBox objective [**erhan2014scalable**]. But it should be extended to handle multiple object categories. Let $x_{ij}^p = \{1, 0\}$ be an indicator for matching the i-th default box to the j-th ground truth box of category p.

The matching strategy above illustrates $\sum_i x_{ij}^p \geq 1$. The overall objective loss function is a weighted sum of the localization loss (loc) and the confidence loss (conf):

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{0.1.1}$$

where N is the number of matched default boxes. If $N = 0$, loss equals to 0. The localization loss is a Smooth L1 loss between the predicted box (l) and the ground truth box (g) parameters. The remaining is similar to Faster R-CNN [**ren2015faster**].

$$\begin{aligned} L_{loc}(x, l, g) &= \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m) \\ \hat{g}_j^{cx} &= (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h \\ \hat{g}_j^w &= \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right) \end{aligned} \tag{0.1.2}$$

The confidence loss is the softmax loss over multiple classes confidences (c).

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^p log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \tag{0.1.3}$$

and the weight term α is set to 1 by cross validation.