

参考文献: yolo-v3和SSD的一些对比

	SSD	YOLOv3
Loss	Softmax loss、Smooth L1 Loss	Logistic loss、回归损失和yolov1的loss
Feature extractor	VGG19(有改写)	Darknet-53
Bounding Box Prediction	direct offset with default box	offset with grid cell by sigmoid
Anchor box	Different scale and aspect ratio	K-means from coco and VOC
Small objects	Semantic value for bottom layer is not high. Worse for small objects.	Higher resolution layers have higher semantic values. Better for small objects.
Big objects	Better. Feature map ranges from $38 * 38$ to $3 * 3, 1 * 1$ .	Worse. $13 * 13$ feature map is the lowest resolution.
Data Augmentation	different sample IOU crop on original image	randomly put the scaled original image on the gray canvas
Input	resize original image to fixed size	Random multi-scale input

- Bounding Box Prediction那里是 $grid\_offset + sigmoid(offset)$
- Big Objects 那里是特征图取得shape越小大目标检测越好
- 就单纯从网络来看, darknet在吸收了残差网络的优点之后, 应该比vgg能力强一些
- 在anchor boxes方面SSD固定8732个anchor boxes, 而YOLO v3有 $52 * 52 * 3$ 的一层, 而且YOLO v3的输入是300到600间32的倍数, 所以就anchor boxes数量来说应该是YOLO v3多
- 速度上来说yolo自己给出的图是比ssd又快又精确, 但我有些想不通, 可能是因为darknet-53比vgg优秀很多, ssd把vgg网络还改写过(可能是这些原因)

SSD和YOLO可以这么比较, 是因为两个算法的步骤已经非常相似了

特征提取 => anchor boxes => loss

## Faster-RCNN

### 1. 在处理feature尺度问题上

Faster-RCNN 用ROI Pooling来统一proposal的尺度

SSD用Multi Layer来考虑各个尺度

YOLO用不同的输入考虑

### 2. Faster-RCNN多个rpn

这部分是Faster-RCNN的核心部分

### 3. 只提取一个特征层

SSD和YOLOv3都对多个feature map进行提取

### 4. anchor boxes size取法不同

SSD的anchor boxes是算出来的

Faster-RCNN的是固定的

YOLO是用数据集的gt boxes 通过kmeans算出来的

### 5. feature extractor

SSD - VGG

YOLO - Darknet

Faster-RCNN - Inception-Resnet v2

### 6. Faster-RCNN慢的原因

Faster-RCNN在feature map每个像素点取9个anchor boxes

之后proposal数量