SSD(single shot multibox detector)论文笔记

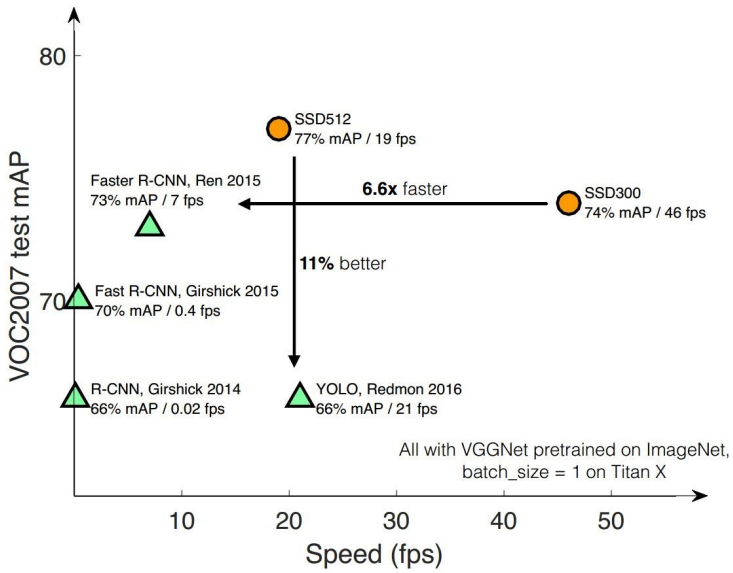
trimmer:**liyingjie**

SSD paper

SSD github

参考1 参考2

- SSD:相比Faster RCNN有明显的速度优势，相比YOLO又有明显的mAP优势(SDD300=INPUT 300 * 300，SDD512=INPUT 512 * 512,不过已经被CVPR 2017的YOLO9000超越）



SSD的主要特点如下：

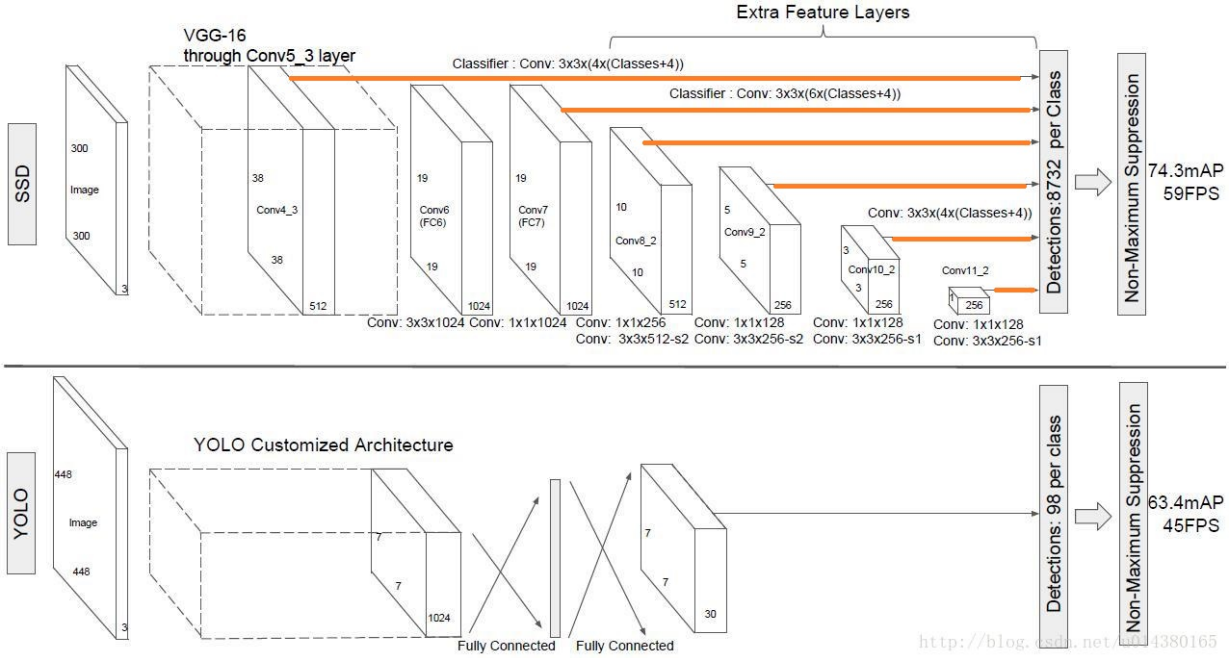
- 1：采用了特征融合，加入基于特征金字塔（Pyramidal Feature Hierarchy）的检测方式，相当于半个FPN思路
- 2：基于Faster RCNN中的anchor，提出了相似的prior box（default box）
- 3：从YOLO中继承了将detection转化为regression的思路，可以进行端到端的网络训练

SSD详解

- 算法的主网络结构是VGG16，将两个全连接层改成卷积层再增加4个卷积层构造网络结构。对其中5个不同的卷积层的输出分别用两个3*3的卷积核进行卷积，一个输出分类用的confidence，每个default box生成21个confidence（这是针对VOC数据集包含20个object类别而言的）；一个输出回归用的localization，每个default box生成4个坐标值（x，y，w，h）。另外这5个卷积层还经过priorBox层生成default box（生成的是坐标）。上面所述的5个卷积层中每一层的default box的数量是给定的。最后将前面三个计算结果分别合并然后传递给loss层。 **作者认为自己的算法之所以在速度上有明显的提升，得益于去掉了bounding box proposal以及后续的pixel或feature的resampling步骤。**

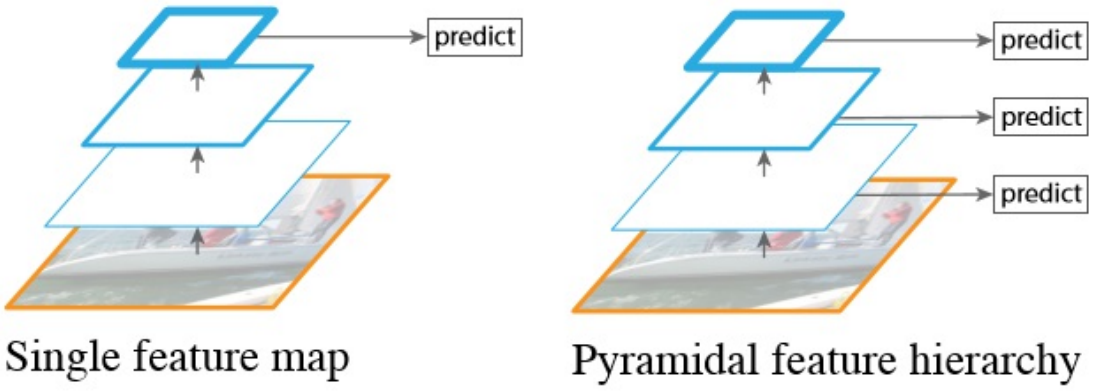
以SSD300为例进行分析

1.网络结构如下：



- 由图可知：

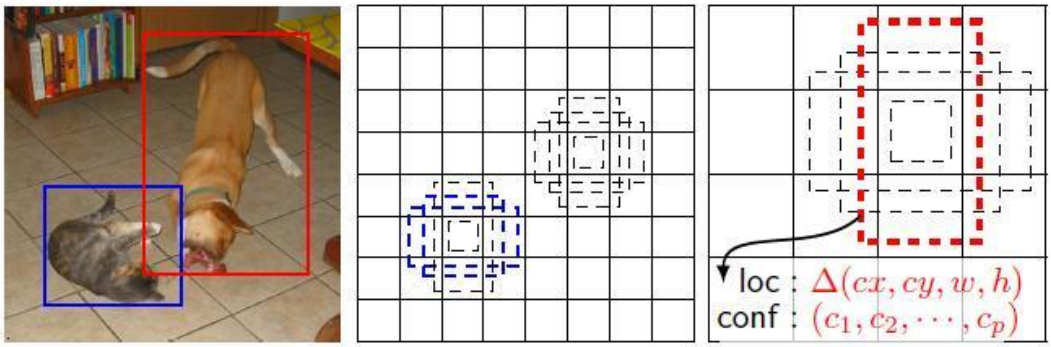
- 可以看到YOLO在卷积层后接全连接层，即检测时只利用了最高层feature maps（包括Faster RCNN也是如此）；YOLO算法的输入是448 * 448 * 3，输出是7 * 7 * 30，这7 * 7个grid cell一共预测98个bounding box。
    - SSD相对YOLO其中容易被忽视的一点: 去掉了YOLO的fc，纯CNN，所以可对任意大小的图片进行识别。
    - **SSD采用了特征金字塔结构进行检测，（在ImageNet数据集上预训练完以后用两个新的卷积层代替fc6和fc7）即检测时利用了conv4-3，conv-7（FC7），conv6-2，conv7-2，conv82，*conv92*这些大小不同的feature maps，在多个feature maps上同时进行softmax分类和位置回归（offset和confidence）。**如下图：



2 Prior Box（default box）：

- **prior box，是指实际中选择的default box（每一个feature map cell 不是k个default box都取）**
- 在SSD中引入了Prior Box，实际上与anchor非常类似，就是一些目标的预选框，后续通过softmax分类+bounding box regression获得真实目标的位置。
    - **default box**：是指在feature map的每个小格(cell)上都有一系列固定大小的box，如下图，有8 * 8和4 * 4两种大小的feature maps，而feature map cell就是其中的每一个小格。图中每个小格子有4个default box。假设每个feature map cell有k个default box，那么对于每个default box都需要预测c个类别score和4个offset，那么如果一个feature map的大小是m * n，也就是有m * n个feature map cell，那么这个feature map就一共有k * m * n个box。
- 实验表明default box的shape数量越多，效果越好。 所以这里用到的default box和Faster RCNN中的anchor很像，**在Faster RCNN中anchor只用在最后一个卷积层，但是在本文中，default box是应用在多个不同层的feature map上。**
- **在训练阶段**，算法在一开始会先将这些default box和ground truth box进行匹配，比如蓝色的两个虚线框和猫的ground truth box匹配上了，一个红色的虚线框和狗的ground truth box匹配上了。所以一个ground truth可能对应多个default box。在预测阶段，直接预测每个default box的偏移

以及对每个类别相应的得分，最后通过NMS得到最终的结果。



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Fig. 1: **SSD framework.** (a) SSD only needs an input image and ground truth boxes for each object during training. In a convolutional fashion, we evaluate a small set (e.g. 4) of default boxes of different aspect ratios at each location in several feature maps with different scales (e.g. 8×8 and 4×4 in (b) and (c)). For each default box, we predict both the shape offsets and the confidences for all object categories ($(c_1, c_2, \cdots, c_p)$). 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).

default box的scale（大小）和aspect ratio（横纵比）：

- 假设我们用m个feature maps做预测，那么对于每个featuer map而言其default box的scale是按以下公式计算的：

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), \quad k \in [1, m]$$

这里smin是0.2，表示最底层的scale是0.2,；smax是0.9，表示最高层的scale是0.9。至于aspect ratio，用ar表示为下式：注意这里一共有5种aspect ratio :

$$a_r = \{1, 2, 3, \tfrac{1}{2}, \tfrac{1}{3}\}$$

因此每个default box的宽的计算公式为：

$$w_k^a = s_k \sqrt{a_r}$$

高：$h_k^a = s_k / \sqrt{a_r}$

另外当aspect ratio为1时，作者还增加一种scale的default box：

$$s_k' = \sqrt{s_k s_{k+1}}$$

- 对于每个feature map cell而言，一共有6种default box。可以看出这种default box在不同的feature层有不同的scale，在同一个feature层又有不同的aspect ratio，因此基本上可以覆盖输入图像中的各种形状和大小的object！

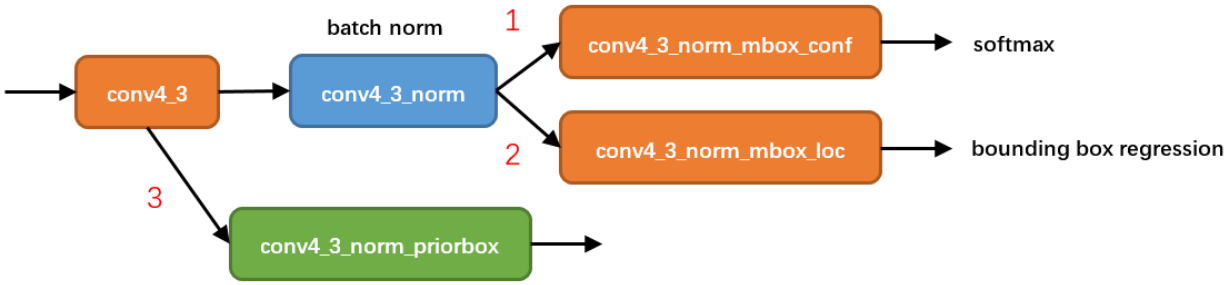
第一层feature map对应的min*size=S1*，*max*size=S2；第二层min*size=S2*，*max*size=S3；其他类推。在原文中，Smin=0.2，Smax=0.9，但是在SSD 300中prior box设置并不能和paper中上述公式对应：

	min_size	max_size
conv4_3	30	60
fc7	60	111
conv6_2	111	162
conv7_2	162	213
conv8_2	213	264
conv9_2	264	315

- SSD使用低层feature map检测小目标，使用高层feature map检测大目标，这也应该是SSD的突出贡献了。

prior box的使用

- 以conv4_3为例进行分析。



-

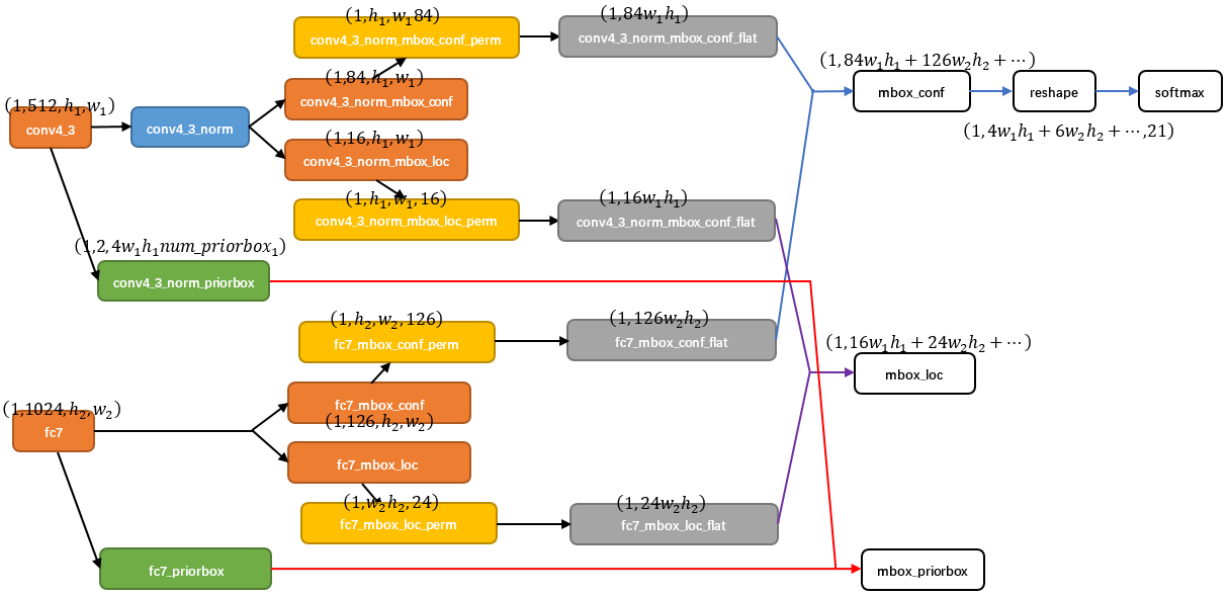
  在conv4_3 feature map网络pipeline分为了3条线路：

  - 经过一次batch norm+一次卷积后，num*class* * *num*priorbox大小的feature用于softmax分类目标和非目标 （num*class是目标类别，SSD 300中num*class = 21）
  - 经过一次batch norm+一次卷积后，生成了4 * num_priorbox,大小的feature用于bounding box regression（即每个点一组[dxmin，dymin，dxmax，dymax]），
  - 生成了[2,4 * num_priorbox]大小的prior box blob，其中2个channel分别存储prior box的4个点坐标和对应的4个variance(变量variance用来对bbox的回归目标进行放大，从而加快对应滤波器参数的收敛。)

后续通过softmax分类+bounding box regression即可从priox box中预测到目标

多个feature maps协同工作

- **将不同size的feature map组合在一起进行prediction。下图仅展示了conv4_3和fc7合并在一起的过程。**



- 对于conv4 _ 3 feature map，（conv4 _ 3 _ norm _ priorbox）（priorbox层）设置了每个点共有4个prior box。由于SSD 300共有21个分类，所以（conv4 _ 3_ norm _ mbox _ conf）的channel值为num _ priorbox * num_class = 4 * 21 = 84；而每个prior box都要回归出4个位置变换量，所以conv4 _ 3 _ norm _ mbox _ loc的caffe blob channel值为4 * 4 = 16。
- fc7每个点有6个prior box，其他feature map同理。
- 经过一系列图7展示的caffe blob shape变化后，最后拼接成mbox*conf*和*mbox*loc。而mbox_conf后接reshape，再进行softmax（为何在softmax前进行reshape，Faster RCNN有提及）。
- 最后这些值输出detection*out*layer，获得检测结果

SSD的不足：

- 1：需要人工设置prior box的min_size，max*size*和*aspect*ratio值。网络中prior box的基础大小和形状不能直接通过学习获得，而是需要手工设置。而网络中每一层feature使用的prior box大小和形状恰好都不一样，导致调试过程非常依赖经验。
- 2：虽然采用了pyramdial feature hierarchy的思路，但是对小目标的recall依然一般，并没有达到碾压Faster RCNN的级别。作者认为，这是由于SSD使用conv4_3低级feature去检测小目标，而低级特征卷积层数少，存在特征提取不充分的问题。

训练阶段

- 损失函数：

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

- 对于SSD，虽然paper中指出采用了所谓的“multibox loss”，但是依然可以清晰看到SSD loss分为了confidence loss和location loss两部分，其中N是match到GT（Ground Truth）的prior box数量；而α参数用于调整confidence loss和location loss之间的比例，默认α=1。SSD中的confidence loss是典型的softmax loss：

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

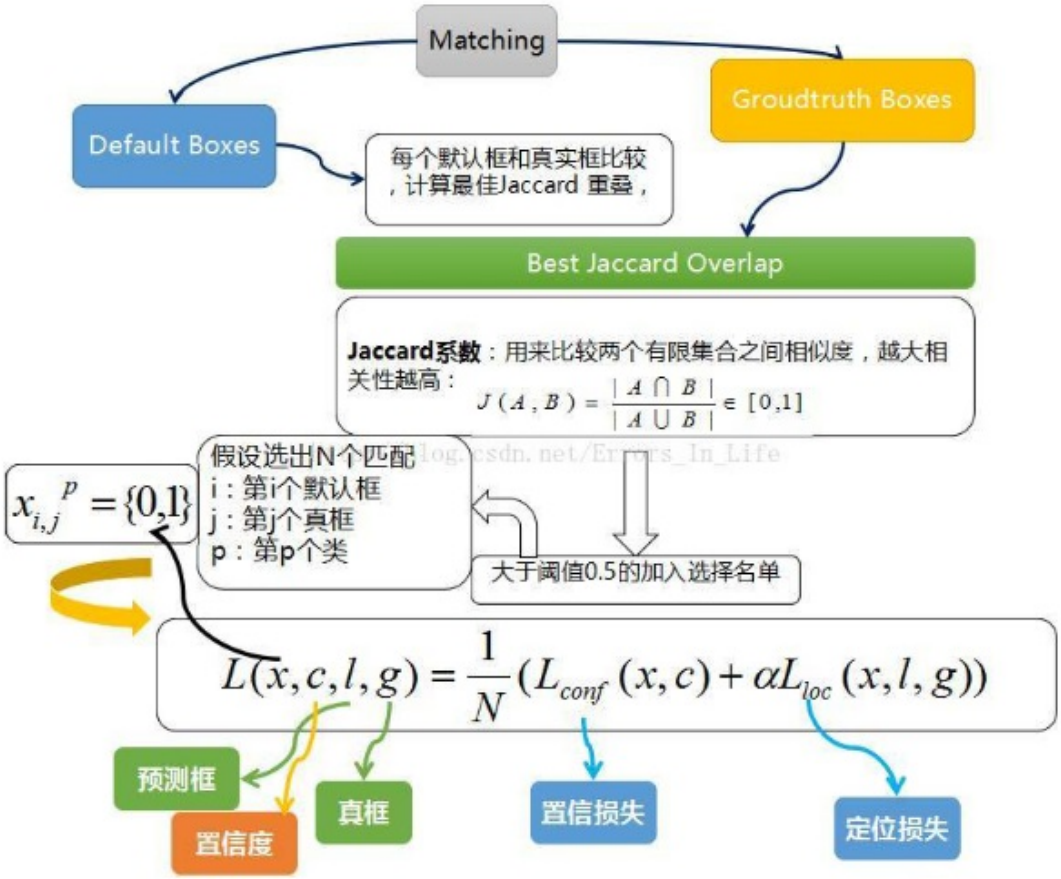
- 其中$x_{ij}^{p}=\{1,0\}$代表第i个prior box匹配到了第j个class为p类别的GT box；而location loss是典型的smooth L1 loss：

$$L_{loc}(x,l,g)=\sum_{i\in Pos}^{N}\sum_{m\in\{cx,cy,w,h\}}x_{ij}^{k}\text{smooth}_{\text{L1}}(l_i^m-\hat{g}_j^m)$$

$$\hat{g}_j^{cx}=(g_j^{cx}-d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy}=(g_j^{cy}-d_i^{cy})/d_i^h$$

$$\hat{g}_j^{w}=\log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^{h}=\log\left(\frac{g_j^h}{d_i^h}\right)$$

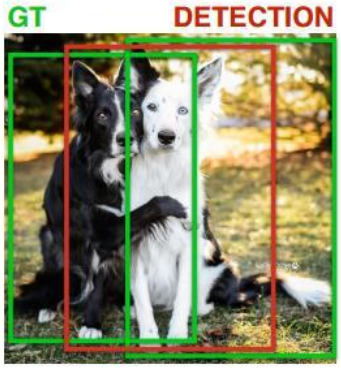
一个很好的参考图：



几种object detection算法的default boxes数量：

Why So Many Default Boxes?

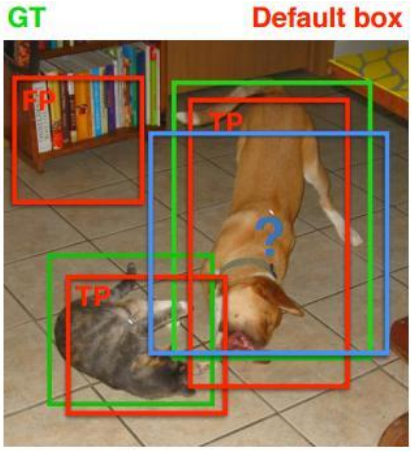
	Faster R-CNN	YOLO	SSD300	SSD512
# Default Boxes	6000	98	8732	24564
Resolution	1000x600	448x448	300x300	512x512



- SmoothL1 or L2 loss for box shape averages among likely hypotheses
- Need to have enough default boxes (discrete bins) to do accurate regression in each
- General principle for regressing complex continuous outputs with deep nets

Matching strategy:

- Matching ground truth and default boxes
  - Match each GT box to closest default box
  - Also match each GT box to all unassigned default boxes with IoU > 0.5
- Hard negative mining
  - Unbalanced training: 1-30 TP, 8k-25k FP
  - Keep TP:FP ratio fixed (1:3), use worst-misclassified FPs.



- **注意：一般情况下negative default boxes数量>>positive default boxes数量，直接训练会导致网络过于重视负样本，从而loss不稳定。所以SSD在训练时会依据confidience score排序default box，挑选其中confidience高的box进行训练，控制positive：negative=1：3**

Data augmentation:

- **数据增广，即每一张训练图像，随机的进行如下几种选择：**
  - **使用原始的图像**
  - **采样一个 patch，与物体之间最小的 jaccard overlap 为：0.1，0.3，0.5，0.7 或 0.9**
  - **随机的采样一个 patch**
- **采样的 patch 是原始图像大小比例是[0.1，1]，aspect ratio在1/2与2之间。当 groundtruth box 的 中心（center）在采样的patch中时，保留重叠部分。在这些采样步骤之后，每一个采样的patch被resize到固定**

的大小，并且以0.5的概率随机的 水平翻转（horizontally flipped）。

其实Matching strategy，Hard negative mining，Data augmentation，都是为了加快网络收敛而设计的。尤其是Data augmentation，翻来覆去的randomly crop，保证每一个prior box都获得充分训练而已。不过当数据达到一定量的时候，不建议再进行Data augmentation，毕竟“真”的数据比“假”数据还是要好很多。