# SSD网络

SSD是一种非常优秀的one-stage目标检测方法，one-stage算法就是目标检测和分类是同时完成的，其主要思路是利用CNN提取特征后，均匀地在图片的不同位置进行密集抽样，抽样时可以采用不同尺度和长宽比，物体分类与预测框的回归同时进行，整个过程只需要一步，所以其优势是速度快。

但是均匀的密集采样的一个重要缺点是训练比较困难，这主要是因为正样本与负样本（背景）极其不均衡（参见Focal Loss），导致模型准确度稍低。

SSD的英文全名是Single Shot MultiBox Detector，Single shot说明SSD算法属于one-stage方法，MultiBox说明SSD算法基于多框预测。

## 1.1 VGG-16网络结构

在讨论SSD之前，先说一下，VGG-16，16个卷积层和3个全连接层。如下图1.1所示。

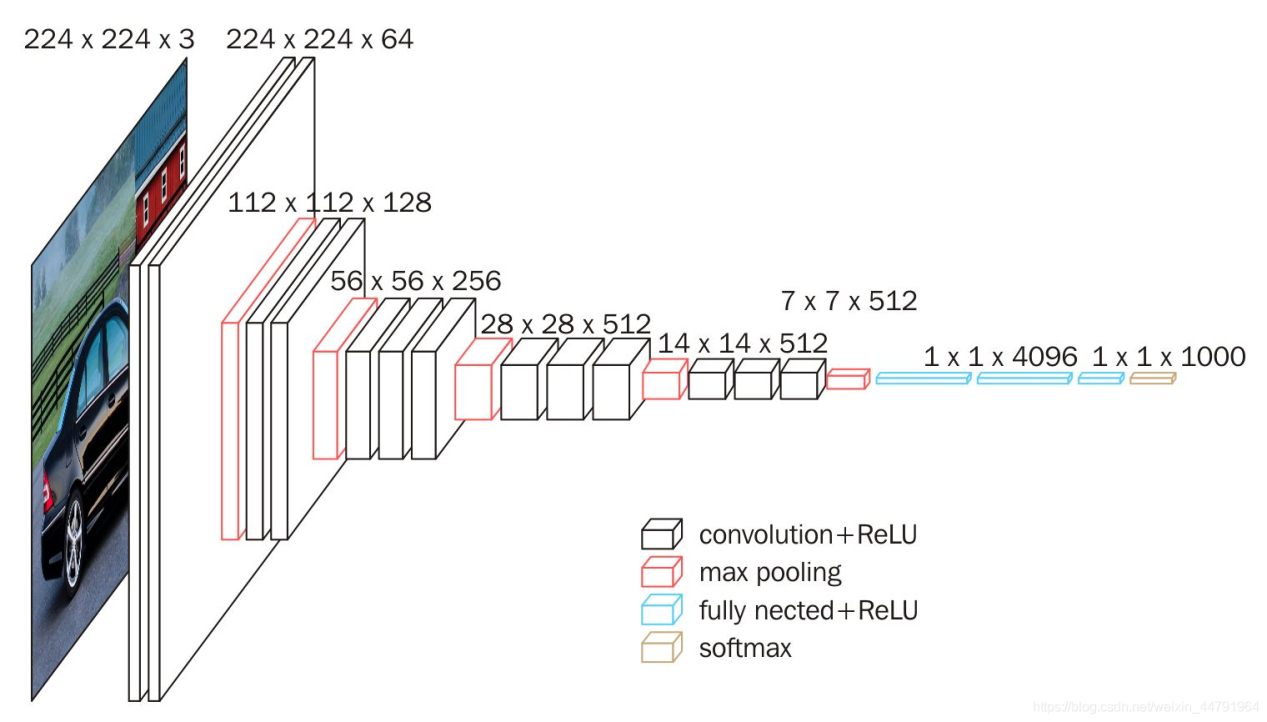


图1.1 VGG16示意图

该模型参加2014年的 ImageNet图像分类与定位挑战赛，取得了优异成绩：在分类任务上排名第二，在定位任务上排名第一。

上图1.1将VGG-16的结构描述的很清楚，下面也会详细的说一下网络的

* 一张原始图片被resize到(224,224,3)。
* conv1两次[3,3]卷积网络，输出的特征层为64，输出为(224,224,64)，再2X2最大池化，输出net为(112,112,64)。
* conv2两次[3,3]卷积网络，输出的特征层为128，输出net为(112,112,128)，再2X2最大池化，输出net为(56,56,128)。
* conv3三次[3,3]卷积网络，输出的特征层为256，输出net为(56,56,256)，再2X2最大池化，输出net为(28,28,256)。
* conv4三次[3,3]卷积网络，输出的特征层为512，输出net为(28,28,512)，再2X2最大池化，输出net为(14,14,512)。
* conv5三次[3,3]卷积网络，输出的特征层为512，输出net为(14,14,512)，再2X2最大池化，输出net为(7,7,512)。
* 利用卷积的方式模拟全连接层，效果等同，输出net为(1,1,4096)。共进行两次。
* 利用卷积的方式模拟全连接层，效果等同，输出net为(1,1,1000)。
* 最后输出的就是每个类的预测。

## 2. SSD网络

如图2.1所示，SSD网络首先将图片resize到300\*300\*3，为防止失真，在图片周围补充灰条。图像多次卷积和压缩后，小物体的信息容易丢失，所以38\*38,19\*19的网格用于检测大物体，而小尺寸的网格来检测大物体，这个也和感受野相关，小尺寸的feature map感受野比较大，用来检测大物体，而大尺寸的feature map用来检测大物体。



SSD采用的主干网络是VGG网络，下面会继续说一下SSD网络，总体来讲，SSD网络的结构就是在VGG-16的基础上改进得来的，如下图2.2所示：

* + 将VGG16的FC6和FC7层转化为卷积层；
  + 去掉所有的Dropout层和FC8层；
  + 新增了Conv6、Conv7、Conv8、Conv9。

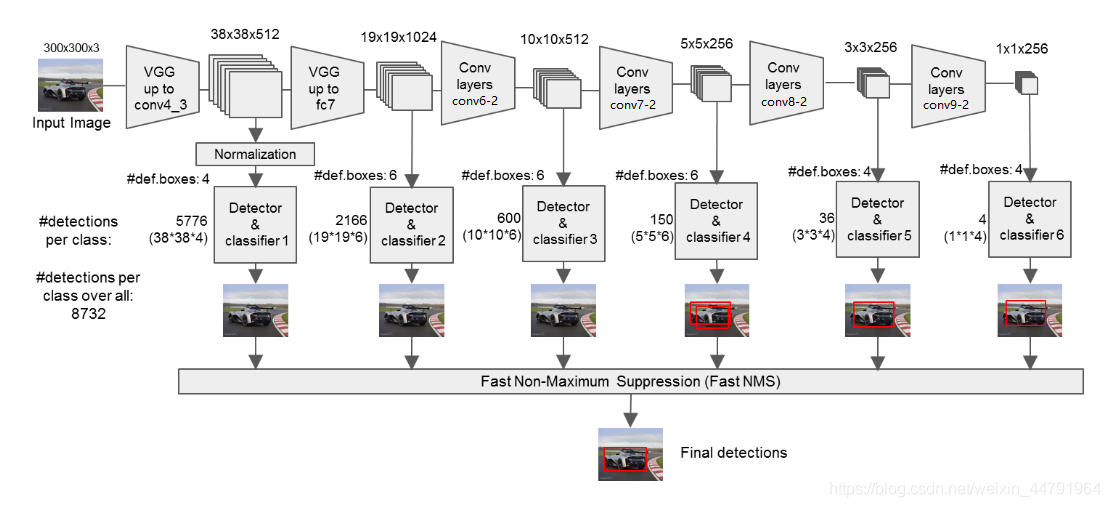


图2.2 SSD网络结构图

下图2.3是SSD的网络的具体流程图。

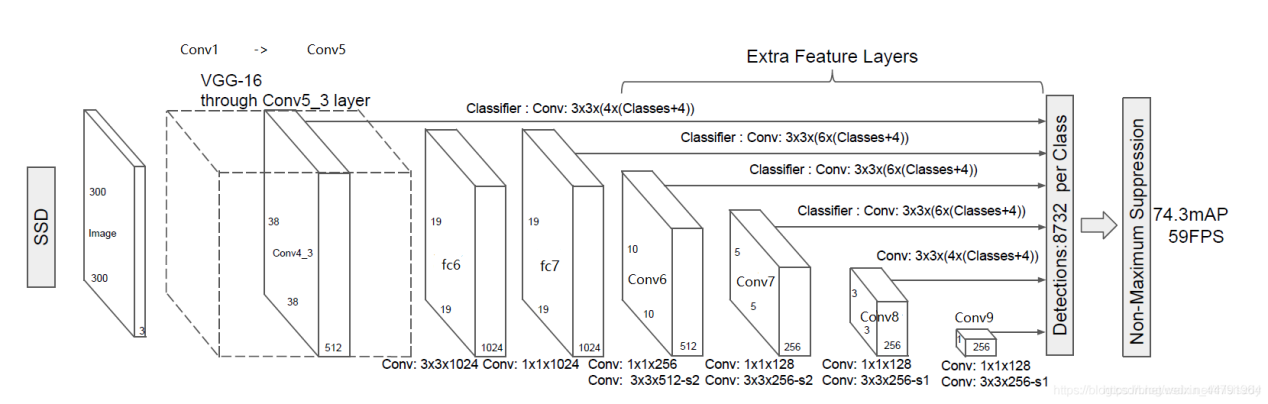


图2.3 SSD网络具体流程图

如图所示，输入的图片经过了改进的VGG网络（Conv1->fc7）和几个另加的卷积层（Conv6->Conv9），进行特征提取：

1. 输入一张图片后，被resize到300x300的shape；
2. conv1，经过两次[3,3]卷积网络，输出的特征层为64，输出为(300,300,64)，再2X2最大池化，该最大池化步长为2，输出net为(150,150,64)；
3. conv2，经过两次[3,3]卷积网络，输出的特征层为128，输出net为(150,150,128)，再2X2最大池化，该最大池化步长为2，输出net为(75,75,128)；
4. conv3，经过三次[3,3]卷积网络，输出的特征层为256，输出net为(75,75,256)，再2X2最大池化，该最大池化步长为2，输出net为(38,38,256)；
5. conv4，经过三次[3,3]卷积网络，输出的特征层为512，输出net为(38,38,512)，再2X2最大池化，该最大池化步长为2，输出net为(19,19,512)；
6. conv5，经过三次[3,3]卷积网络，输出的特征层为512，输出net为(19,19,512)，再3X3最大池化，该最大池化步长为1，输出net为(19,19,512)；
7. 利用卷积代替全连接层，进行了一次[3,3]卷积网络和一次[1,1]卷积网络，分别为fc6和fc7，输出的通道数为1024，因此输出的net为(19,19,1024)。（从这里往前都是VGG的结构）；
8. conv6，经过一次[1,1]卷积网络，调整通道数，一次步长为2的[3,3]卷积网络，输出的通道数为512，因此输出的net为(10,10,512)；
9. conv7，经过一次[1,1]卷积网络，调整通道数，一次步长为2的[3,3]卷积网络，输出的通道数为256，因此输出的net为(5,5,256)；
10. conv8，经过一次[1,1]卷积网络，调整通道数，一次padding为valid的[3,3]卷积网络，输出的通道数为256，因此输出的net为(3,3,256)；
11. conv9，经过一次[1,1]卷积网络，调整通道数，一次padding为valid的[3,3]卷积网络，输出的特征层为256，因此输出的net为(1,1,256)；

### 2.1 多尺度特征映射

  如图2.4所示，左边的方法针对输入的图片获取不同尺度的特征映射，但是在预测阶段仅仅使用了最后一层的特征映射；而SSD（右图）不仅获得不同尺度的特征映射，同时在不同的特征映射上面进行预测，它在增加运算量的同时可能会提高检测的精度，因为它具有更多的可能性。

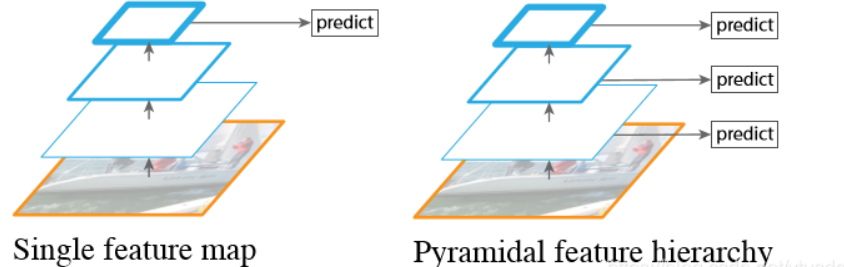
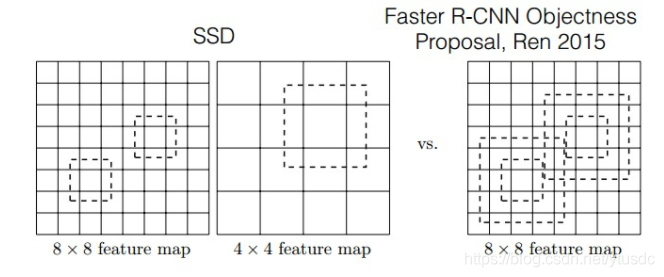


图2.4 单层feature map预测和多层特征金字塔预测对比

如图2.5所示，对于BB（bounding boxes）的生成，Faster-rcnn和SSD有不同的策略，但是都是为了同一个目的，产生不同尺度，不同形状的BB，用来检测物体。对于Faster-rcnn而言，其在特定层的（最后一层）Feature map上面的每一点生成9个预定义好的BB，然后进行回归和分类操作来进行初步检测，然后进行ROI Pooling和检测获得相应的BB；而SSD则在不同的特征层的feature map上的每个点同时获取6个（有的层是4个）不同的BB，然后将这些BB结合起来，最后经过NMS（非极大值抑制）处理获得最后的BB

图2.5 Anochors 生成比较

如下图2.6所示，我们将猫和狗的照片输入到一个卷积神经网络中，在这期间，经历了多个卷积层和池化层，在不同的卷积层会输出不同大小的feature map（这是由于pooling层的存在，它会将图片的尺寸变小），而且不同的feature map中含有不同的特征，而不同的特征可能对我们的检测有不同的作用。总的来说，浅层卷积层对边缘更加感兴趣，可以获得一些细节信息（位置信息），而深层网络对由浅层特征构成的复杂特征更感兴趣，可以获得一些语义信息，对于检测任务而言，一幅图像中的目标有复杂的有简单的，对于简单的patch我们利用浅层网络的特征就可以将其检测出来，对于复杂的patch我们利用深层网络的特征就可以将其检测出来，因此，如果我们同时在不同的feature map上面进行目标检测，理论上面应该会获得更好的检测效果。

图2.6中，浅层的特征层细节丢失较少，映射到原图上时，区域较小，用来检测小目标，也就是猫；而深层特征层丢失细节比较多，映射到原图上时比较大，用来检测大目标，即原图中的狗。

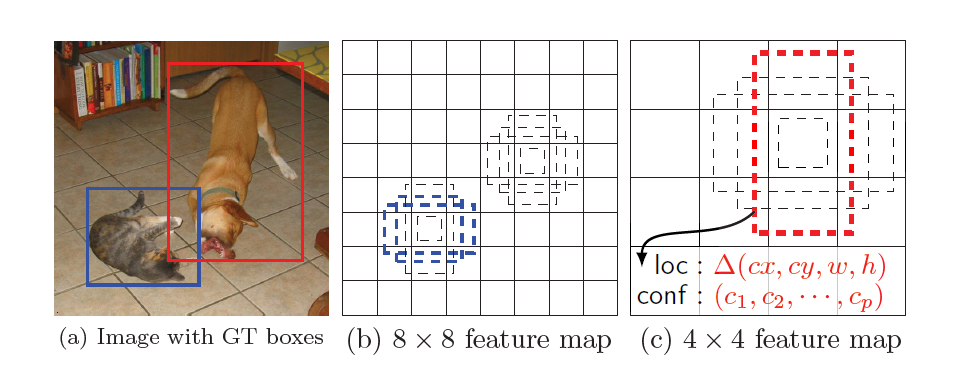


图2.6 浅层特征层与深层特征层

关于default box的选取和尺寸定义，原文中有这么一段文字，如下图2.7所示：

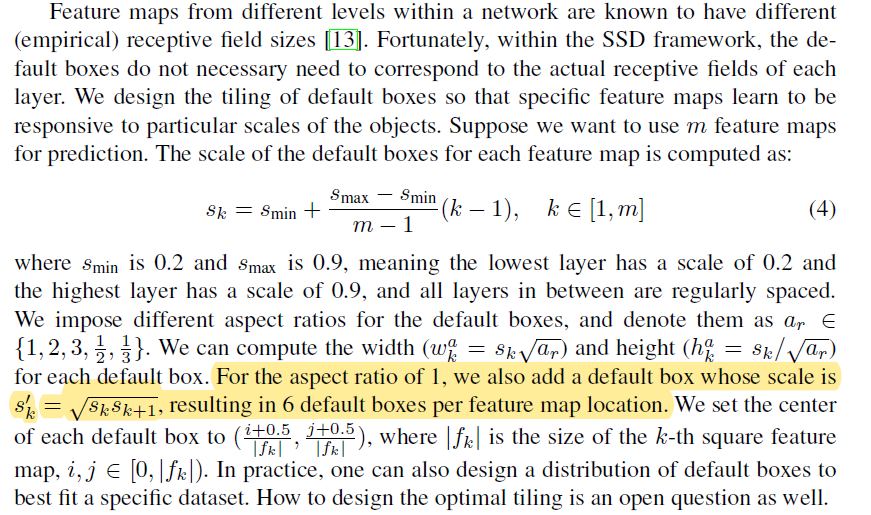
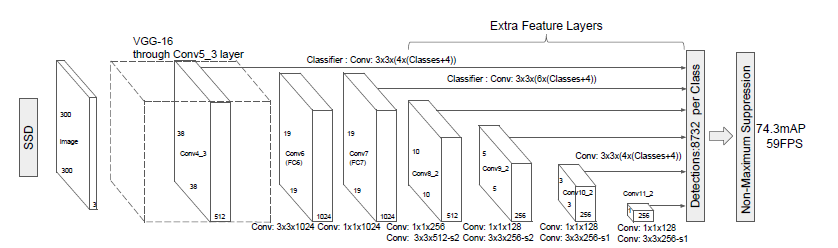


图2.7 原文截图计算default box尺寸的公式

但是，如果根据文中的公式计算，这里直接给出不同特征层上不同尺寸的default的box尺寸和对应的box个数，如表2.1。

表2.1 Default box的Scal 和Aspect ratio设定

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layers of the feature maps | Feature layer size and channl | Scal | Aspect ratio | Defaults boxs | Sum\_box |
| First:  Conv4\_3 | 38\*38\*1024 | (21, 45) | (1, 2, .5) | 38\*38\*4=5776 | 8732 |
| Second:  Fc7 (conv7) | 19\*19\*1024 | (45, 99) | (1, 2,.5,3,1./3) | 19\*19\*6=2166 |
| Third: Conv8\_2 | 10\*10\*256 | (99, 153) | (1, 2,.5,3,1./3) | 10\*10\*6=600 |
| Fourth: Conv9\_2 | 5\*5\*256 | (153, 207) | (1, 2,.5,3,1./3) | 5\*5\*6=150 |
| Fifth: Conv10\_2 | 3\*3\*128 | (207, 261) | (1, 2, .5) | 3\*3\*4=36 |
| Sixth: Conv11\_2 | 1\*1\*128 | (261, 315) | (1, 2, .5) | 1\*1\*4=4 |



这类急需要解释一下，为什么会有两个尺寸，这是因为原文中有公式如下：，原文也有说明，如图2.7中公式所述，对于每一个特征层的1:1的情况，额外增加一个scal即，其中，就是每一层的第一个尺寸，而就是每一层的第二个尺寸，如（21，45），，，即。这样，对于第一层（21,45）的scal下，每点共有4个default box， 因为第一，五，六层的特征层没有3，1./3这两个aspect ratio。第二，三，四层，每个特征层上的每个点有6个default boxes。

图2.7进一步解释了default boxes的设定。

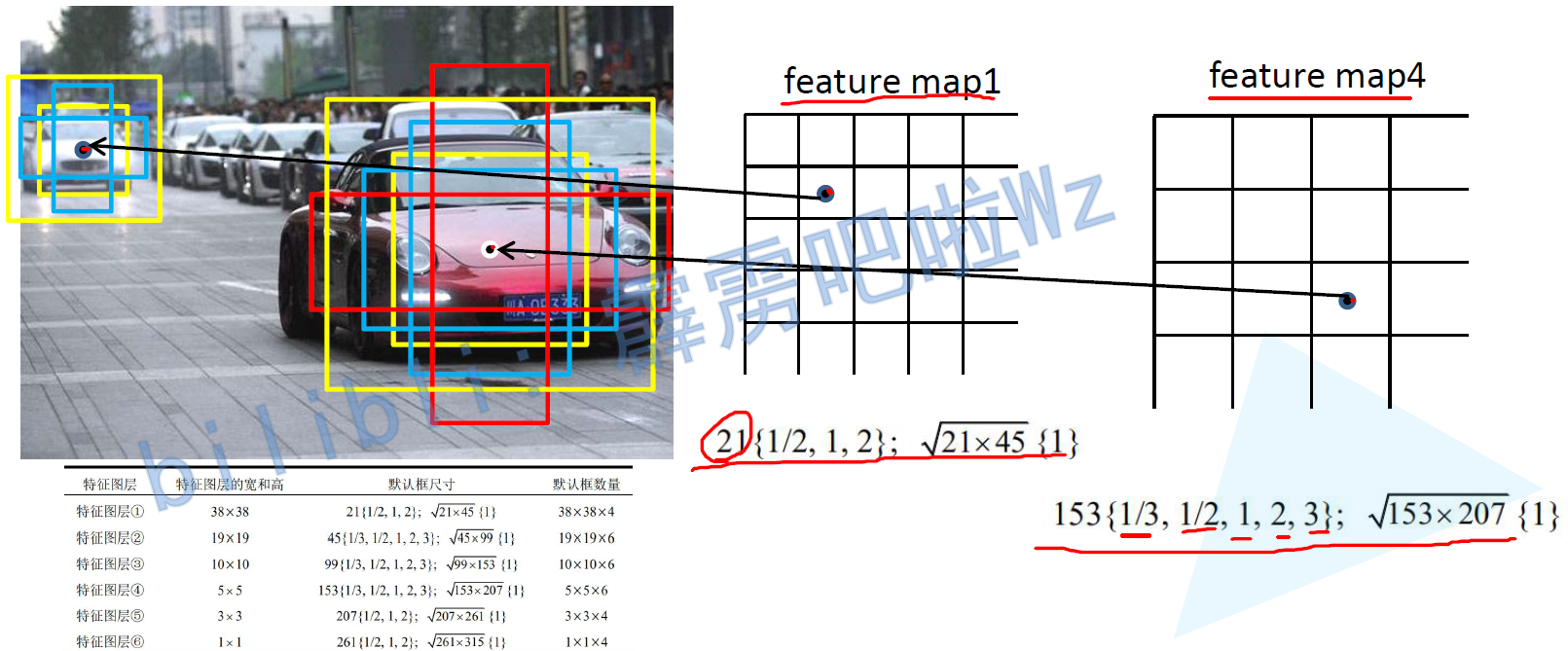


图2.7 default boxes的scal和aspect ratio的设定

2.1.2 Matching strategy（匹配策略）：

在训练时，ground truth boxes 与 default boxe（就是prior boxes） 按照如下方式进行配对：

（1）第一个原则：从ground truth box出发，寻找与每一个ground truth box有最大的jaccard overlap（其实就是IoU）的default box，这样就能保证每一个groundtruth box一定与一个default box对应起来（所谓的jaccard overlap就是IOU，如图12）。 反之，若一个prior box没有与任何ground truth进行匹配，那么该prior box只能与背景匹配，就是负样本。一个图片中ground truth是非常少的， 而default box却很多，如果仅按第一个原则匹配，很多default box会是负样本，正负样本极其不平衡，所以需要第二个原则。

（2）第二个原则：从default box出发，对剩余的还没有配对的default box与任意一个ground truth box尝试配对，只要两者之间的jaccard overlap大于阈值（一般是0.5），那么该default box也与这个ground truth进行匹配。这意味着某个ground truth可能与多个Prior box匹配，这是可以的。但是反过来却不可以，因为一个prior box只能匹配一个ground truth，如果多个ground truth与某个prior box的 IOU 大于阈值，那么groud truth只与IOU最大的那个Prior box进行匹配。注意：第二个原则一定在第一个原则之后进行，仔细考虑一下这种情况，如果某个ground truth所对应最大 IOU小于阈值，并且所匹配的Prior box却与另外一个ground truth的 IOU大于阈值，那么该Prior box应该匹配谁，答案应该是前者，首先要确保某个ground truth一定有一个Prior box与之匹配。但是，这种情况我觉得基本上是不存在的。由于Prior box很多，某个ground truth的最大 IOU 肯定大于阈值，所以可能只实施第二个原则既可以了。

显然配对到GT的default box就是positive，没有配对到GT的default box就是negative。只有正样本才会参与loss的计算。

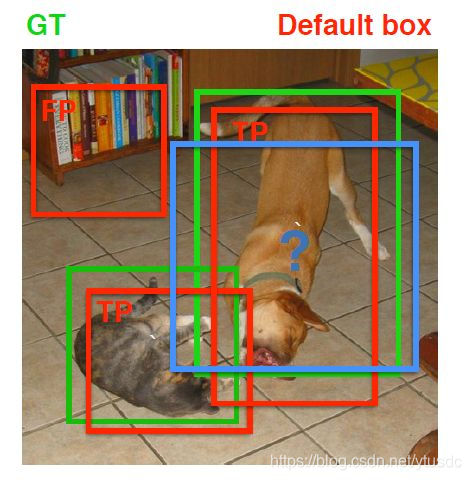


图2.8 匹配示意图

### 2.2 正负样本与Hard negative mining

值得注意的是，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。为了保证正负样本尽量平衡，SSD在训练时采用了hard negative mining，即依据confidience loss对default box进行排序，挑选其中confidience loss高的box进行训练，将正负样本的比例控制在positive：negative=1:3。

#### 2.1 正负样本区分

正负样本区分：（通过定位信息（iou）初步划分，而后通过loss把hard example留下，loss越大越hard）。

#### 2.1.1 获取正样本框

已经确定 default box，结合 ground truth，下一步要将 default box 匹配到 ground truth 上，根据上面两个原则，从 groudtruth box 出发给每个 groudtruth box 找到了最近（IOU最大）的 default box 放入候选正样本集。然后再从 default box 出发为 default box 集中寻找与 groundtruth box 满足IOU>0.5 的default box放入候选正样本集。

#### 2.1.2负样本获得（这是一个难分样本挖掘的过程）

在生成 prior boxes 之后，会产生很多个符合 ground truth box 的 positive boxes（候选正样本集），但同时，不符合 ground truth boxes 也很多，而且这个 negative boxes（候选负样本集），远多于 positive boxes。这会造成 negative boxes、positive boxes 之间的不均衡。训练时难以收敛。

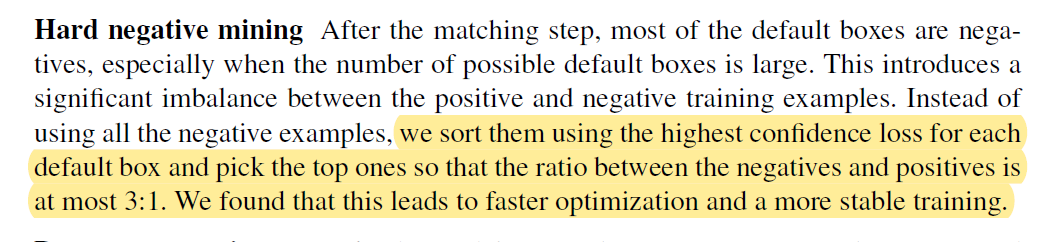
#### 2.1.3 难分数据挖掘

将每一个GT上对应prior boxes的分类loss 进行排序。

对于候选正样本集：选择loss最高的m个 prior box 与候选正样本集匹配 (box 索引同时存在于这两个集合里则匹配成功)，匹配不成功则从候选正样本集中删除这个正样本（之所以选择loss最高的m个prior boxes是因为这个，如果loss太低，意味着prior box与ground truth很接近，容易被识别，所以不在难分样本里，不需要再训练）；

对于候选负样本集：选择loss最高的m个prior box 与候选负样本集匹配，匹配成功的则留下来作为最后的负样本，不成功剔除出候选负样本集。

但是文中讲的是用hand data mining是正负样本比例达到1:3.



#### 2.1.4举例：

假设在这 8732 个 default box 里，经过 FindMatches 后得到候选正样本 P 个，候选负样本那就有 8732−P个。将 prior box 的 prediction loss 按照从大到小顺序排列后选择最高的 M个 prior box。如果这 P 个候选正样本里有 a 个 box 不在这 M 个 prior box 里，则将这 a个 box 从候选正样本集中踢出去。如果这 8732−P个候选负样本集中有 b个在这 M 个 prior box，则将这b个候选负样本作为正式负样本。即删除易识别的正样本，同时留下典型的负样本，组成1:3的prior boxes样本集合。SSD 算法中通过这种方式来保证 positives、negatives 的比例。