1. 了解SSD相关背景

2016 年，Wei Liu，Dragomir Anguelov 等人提出了单射多框检测器方法。

SSD 采用了金字塔结构的特征层，并直接在这些不同尺度上的特征层上通过比较小的 filter 进行物体检测，从而实现不同尺度的物体的检测。

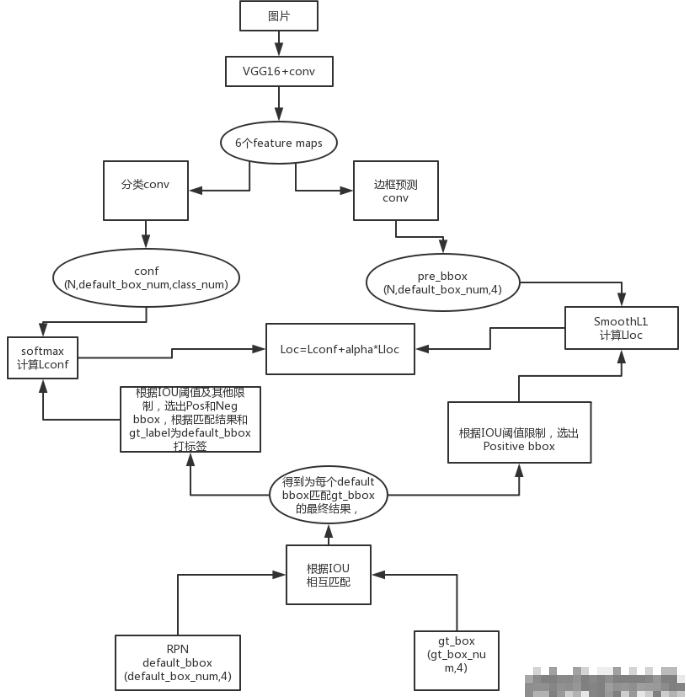
SSD 采用 VGG16 作为特征抽取层，在删掉其尾部的全连接层及输出层后，又添加了 6 层卷积层作为特征扩展。

对于每个特征层，分别采用一个 filter 作为类别预测器（class predictor），另外一个 filter 作为边框预测器（bounding box predictor）。Filter 由小型的卷积神经网络构成，通常是单层的卷积层。当所有的特征层都经过 filters 预测过后，输出结果=类别预测结果+边框预测结果

根据 SSD 的网络结构特征，随着网络越来越深，特征层的分辨率（resolution，或者叫 scale）越来越小。分辨率的下降，所以使SSD 在检测小物体的任务上显得不足。

1. 了解基本组成

网络组成简单结构，在ssd中输入网络的图片是经过augmentation以及其他预处理之后的像素矩阵



网络总体图

1张训练图片上可能只有几个gt\_box，因此需要确定选择那些anchor来负责预测这几个gt\_box

ssd论文中的匹配策略：

所有的gt\_box选择与它iou最大的anchor进行匹配，而且这个iou必须大于设定阈值(0.5)时才算匹配上，匹配上的就是正样本。

1. 和RPN的关系

RPN(Region Proposal Network)可以找到图片中每个物体的种类和位置，如果更注重速度而不是精度的话完全可以只使用RPN。

RPN是一个全卷积网络(FCN)，由于没有全连接层，所以可以输入任意分辨率的图像，可得到一个feature map，利用这个feature map得到物体的位置和类别。

SSD和RPN最大的不同在于RPN只是在最后的feature map上预测检测的结果，而最后一层的feature map往往都比较抽象，对于小物体不能很好地表达特征，而SSD允许从CNN各个level的feature map预测检测结果，这样就能很好地适应不同scale的物体。这就是SSD和RPN最大的不同，其他地方几乎一样。

SSD克服了RPN对小物体检测效果很差的缺点，是对它的改进

SSD与Faster RCNN中的RPN生成anchor的不同之处：

SSD是在多个不同尺度/分辨率的特征图上生成anchor，产生不同宽高比的anchor boxes。

Faster RCNN中的RPN模型，只在一个尺度的特征图上生成anchor。

1. 了解SSD算法

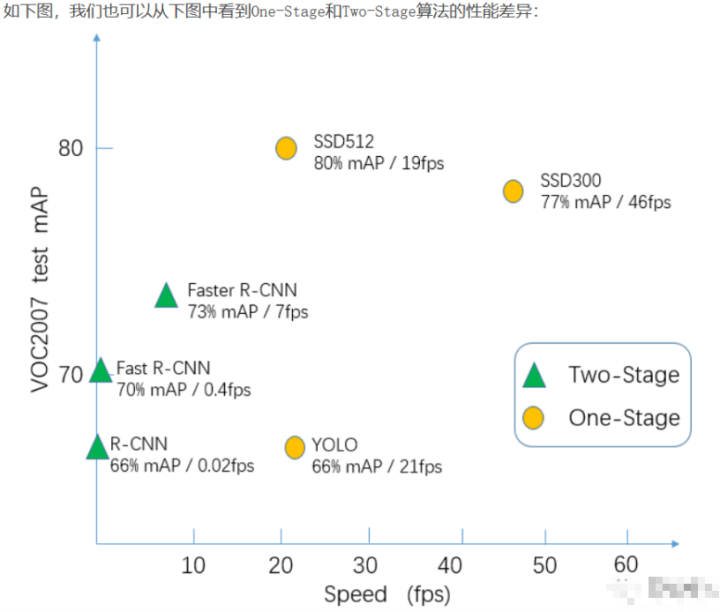
本文提出的SSD算法是一种直接预测目标类别和 bounding box 的多目标检测算法。与 faster rcnn 相比，该算法没有生成 proposal 的过程，这就极大提高了检测速度。

算法的主网络结构是VGG16，将最后两个全连接层改成卷积层，并随后增加了4个卷积层来构造网络结构。

该算法属于One-Stage类算法范畴。

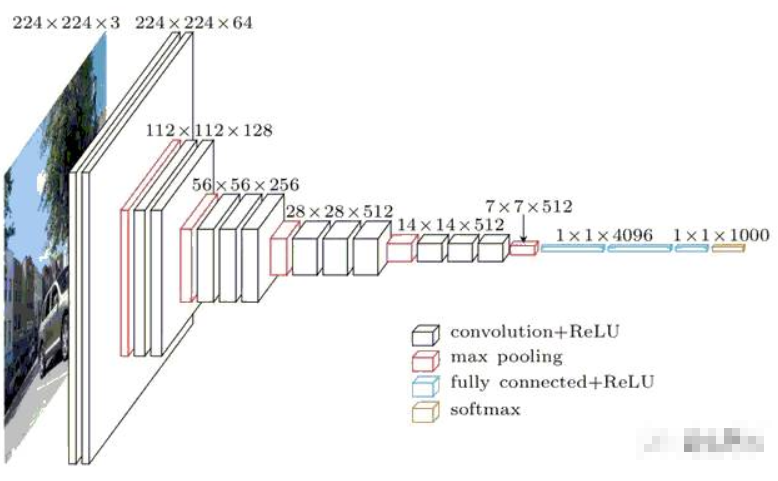
Two-Stage错误率低，漏识别率低，但是速度较慢，不太能满足实时监测场景。

One-Stage：SSD YOLOOne-Stage算法不需要Region Proposals阶段，可以直接产生物体的类别概率和位置坐标值，经过单次检测即可直接得到最终的检测结果。有着更快的检测速度。



可以看到在VOC2007测试集上One-Stage算法有着更快的检测速度。

SSD算法的基础网络是VGG16，我们先看下VGG16的网络结构，如下图：



红色：池化层，黑色：卷积层

VGG16的网络输入是224\*224\*3的图像

经过两个卷积层(Conv1)得到224\*224\*64尺寸的特征图

经过一个池化层和两个卷积层(Conv2)后得到112\*112\*128尺寸的特征图

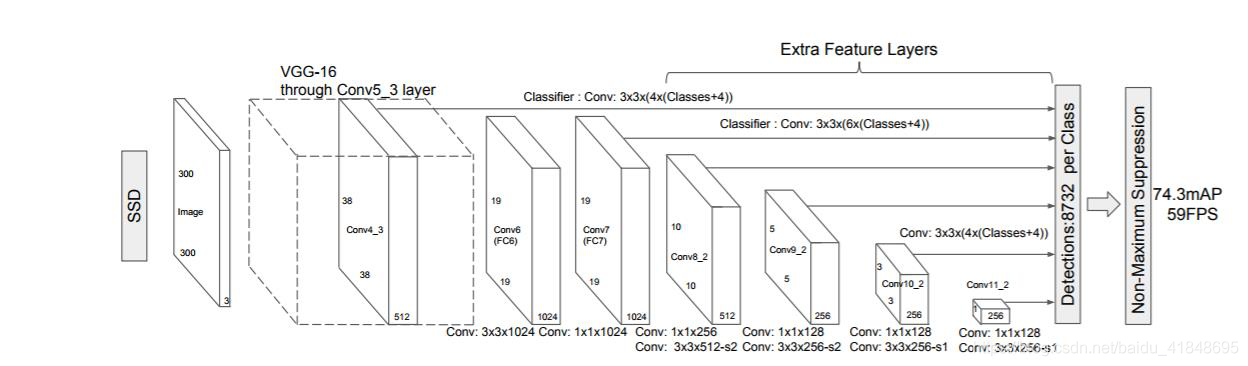
经过一个池化层和三个卷积层后(Conv3)得到56\*56\*256尺寸的特征图

经过一个池化层和三个卷积层后(Conv4)得到28\*28\*512尺寸的特征图

经过一个池化层和三个卷积层后(Conv5)得到14\*14\*512尺寸的特征图

经过一个池化层后得到7\*7\*512尺寸的特征图

最后经过全连接层(FC6&FC7)后再接一个softmax层后输出结果



SSD算法结构流程如下：

1.将尺寸大小为300\*300\*3的图像输入到VGG16网络中，在VGG16的第5步也就是Conv4之后得到38\*38\*512的特征图

2.接着通过一个池化和3\*3\*1024的卷积层Conv6得到19\*19\*1024的特征图

3.之后再经过一个1\*1\*1024的卷积操作Conv7，得到19\*19\*1024的特征图

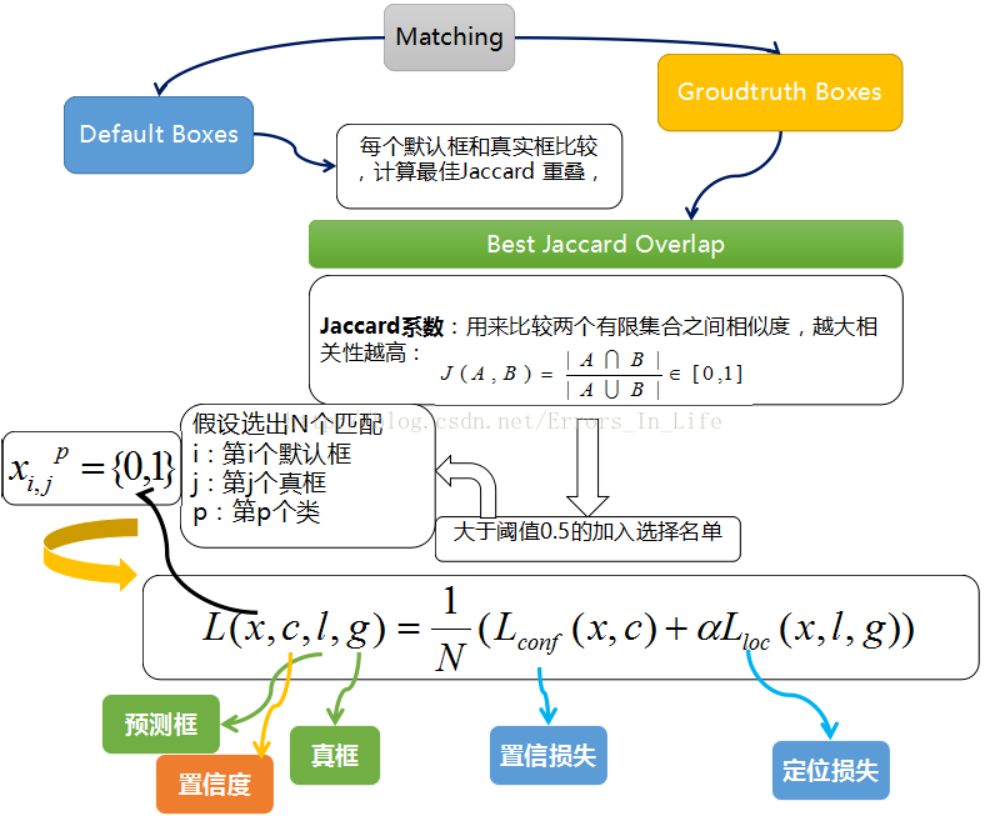
4.接着做一个1\*1\*256和3\*3\*512-s2(s2为stride=2，stride 表示卷积的步长)的卷积操作Conv8后得到10\*10\*512的特征图

5.继续做卷积操作Conv9，具体为1\*1\*128和3\*3\*256-s2卷积之后得到5\*5\*256的特征图

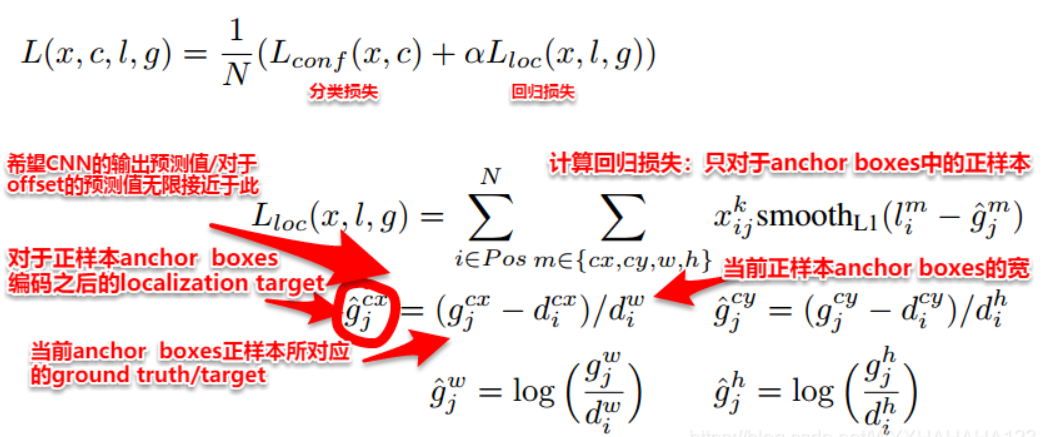
6.继续做卷积操作Conv10，具体为1\*1\*128和3\*3\*256-s1卷积之后得到3\*3\*256的特征图

7.继续做卷积操作Conv11，具体为1\*1\*128和3\*3\*256-s1卷积之后得到1\*1\*256的特征图

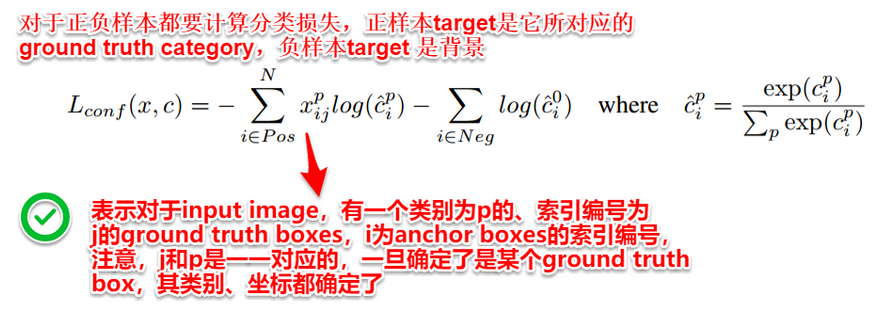
算法公式详细：



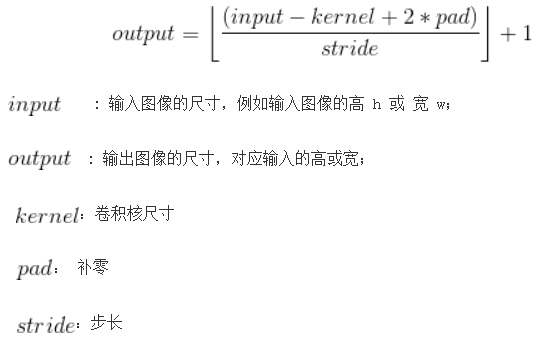
定位损失指的是对目标边界框的坐标回归时产生的误差



置信度损失指的是在对目标做分类回归时产生的误差



1. 卷积补充：

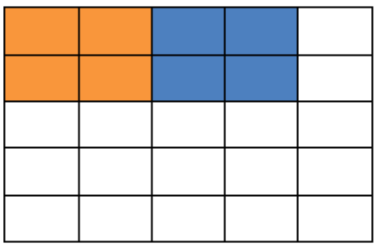


卷积过程中，有时需要通过填充来避免信息损失，有时也要在卷积时通过设置的步长（Stride）来压缩一部分信息。因此卷积中的步幅是另一个构建卷积神经网络的基本操作。

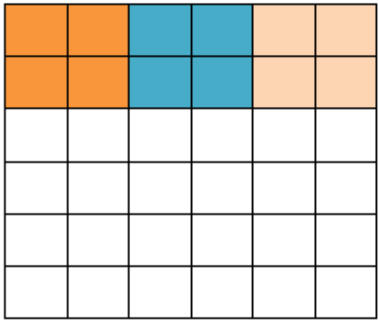
Stride表示filter在原图片中水平方向和垂直方向每次的步进长度。之前我们默认stride=1。若stride=2，则表示filter每次步进长度为2，即隔一点移动一次。

padding的用途: 保持边界信息;可以对有差异的图片进行补齐,使得图像的输入大小一致;在卷积层中加入padding ,会使卷基层的输入维度与输出维度一致; 同时,可以保持边界信息。

以下图为例子，比如有这么一个5\*5的图片（一个格子一个像素），我们滑动窗口取2\*2，步长取2，那么我们发现还剩下1个像素没法滑完，那怎么办呢？



那我们在原先的矩阵加了一层填充值，使得变成6\*6的矩阵，那么窗口就可以刚好把所有像素遍历完。这就是padding的作用。



CNN，它主要用来提取图像特征做分类、识别、预测或决策等。

CNN（卷积神经网络）由卷积层、池化层、全连接层等网络结构组成，通过卷积层提取图像特征，全连接层将特征整合扁平化，最后利用分类器输出图像所属类别的概率，取最大的为物体分类结果。卷积神经网络CNN可以分类+定位。

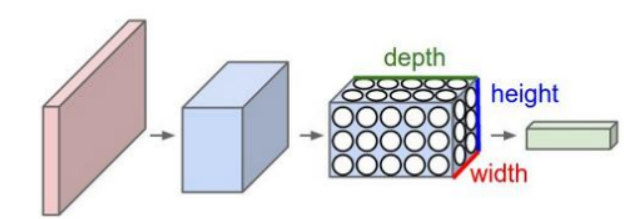
CNN各层作用：

1.卷积层：提取图像特征

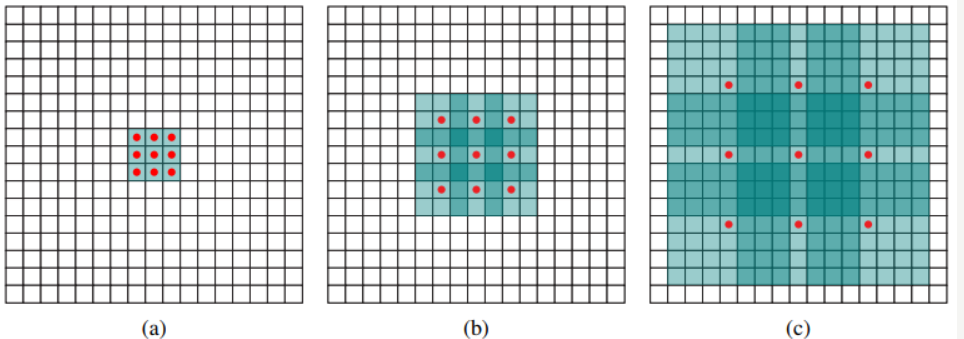
2.池化层：主要用于特征降维，压缩数据和参数的数量，减小过拟合，同时提高模型的容错性。

3.全连接层：输出层，输出特征图片

卷积神经网络是一个多层的神经网络，每层由多个二维平面组成，而每个平面由多个独立神经元组成。

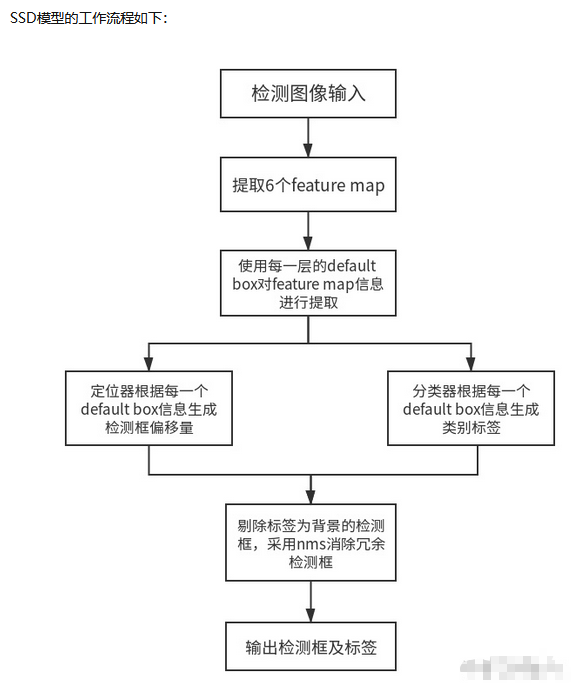


感受野：



1. 是普通的 3\*3 卷积，其视野就是3\*3，(b)是扩张率为2，此时视野变成 7\*7 ，(c)扩张率为4时，视野扩大为 15\*15 ，但是视野的特征更稀疏了。

6.SSD模型工作流程：



7.SSD模型训练流程：

