上海大学无人艇工程研究院

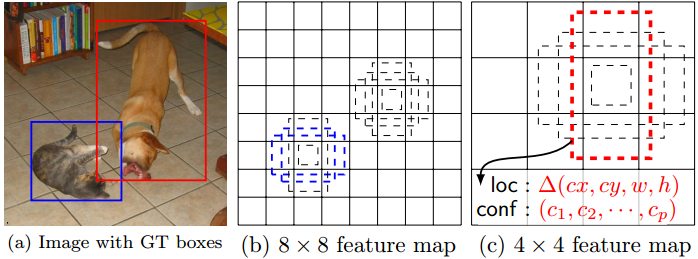
——环境感知组

# SSD: Single Shot MultiBox Detector

论文：SSD: Single Shot MultiBox Detector

作者：**Wei Liu** Dragomir Anguelov Dumitru Erhan Christian Szegedy

出处：2016年ICCV



注：**加粗**的作者为重点关注研究者

图注：本算法的核心示意图

Date：2017.12.05

## 版本更新记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日 期** | **更新人** | **主要更新内容描述** | **版本号** |
| 2017年12月05日 | 陈加宏 | 完成算法大致框架搭建 | V1.0.0 |
| 2017年12月02日 | 陈加宏 | 完成算法细节的总结 | V1.0.1 |

目 录

[SSD: Single Shot MultiBox Detector 1](#_Toc500268126)

[版本更新记录 2](#_Toc500268127)

[1、概述 3](#_Toc500268128)

[1.1 前言 3](#_Toc500268129)

[1.2 创新点 3](#_Toc500268130)

[2、细节 3](#_Toc500268131)

[2.1 主要流程 3](#_Toc500268132)

[2.2 数学模型 3](#_Toc500268133)

[2.3 模型求解 3](#_Toc500268134)

[3、实验 3](#_Toc500268135)

[3.1 代码框架 4](#_Toc500268136)

[3.2 实验结果及分析 4](#_Toc500268137)

[3.3 优缺点总结 4](#_Toc500268138)

[3.4 今后工作 4](#_Toc500268139)

## 1、概述

在目标检测领域，随着深度卷积网络的兴起，出现了两种模式，基于区域检测的RCNN系列算法在检测精度上表现优异，而基于目标回归思路做的YOLO则是在检测实时性方面表现突出，那本文分享的SSD是结合了两者的优势，在保证检测精度的前提下，做到了满足实时要求的目标检测。

### 1.1 前言

基于候选区域和分类的目标检测的方法包括R-CNN系列即R-CNN、SPPnet、Fast R-CNN以及 Faster R-CNN系列，都取得了非常好的结果，但是在速度方面离实时效果还比较远，在提高检测精度的同时兼顾速度，逐渐成为目标检测未来的趋势。YOLO虽然能够达到实时的效果，但是其检测精度与刚才提到的优秀算法的检测结果有很大的差距。并且YOLO有如下一些缺陷：1、每个网格预测一个物体，容易造成漏检；2、对于物体的尺度相对比较敏感，对于尺度变化较大的物体泛化能力较差。针对YOLO中的这些不足，该论文提出的方法 SSD 在这两方面都有所改进，同时兼顾了检测和实时性的要求。在满足实时性的条件下，接近优秀算法的结果。对于输入图像大小调整为300\*300，在VOC2007 test上能够达到58帧每秒，前提是使用Titan X的GPU，72.1%的检测精度。输入图像大小为500 \*500，mAP能够达到75.1%。作者的思路就是Faster R-CNN+YOLO，利用YOLO的思路和Faster R-CNN的anchor box的思想。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithm | DPM | RCNN | Fast RCNN | Faster RCNN | YOLO | SSD |
| Precision/mAP | 34.3 | 58.5 | 70 | 73.2 | 63.4 | 72.1 |
| Speed/FPS | 0.5 | -- | 0.5 | 7 | 45 | 58 |

SSD算法的目标检测流程与YOLO相似，也是直接将图片输入到深度网络中就可以得到图像中目标的位置和置信度，是一种端对端的网络架构的应用。

### 1.2 创新点

本文的主要贡献总结如下：

1、提出了新的物体检测方法：SSD，比原先最快的YOLO: You Only Look Once方法还要快，同时检测更加精确。保证速度的同时，其检测精度可与使用基于区域候选框技术的方法，如Faster R-CNN相媲美；

2、SSD方法的核心就是预测目标类别得分，在特征图上使用小的卷积核，去预测一系列目标框的box offsets；

3、本文中为了得到高精度的检测结果，在不同层次的特征图上去预测目标及其box offsets，还可以得到不同纵横比的预测结果；

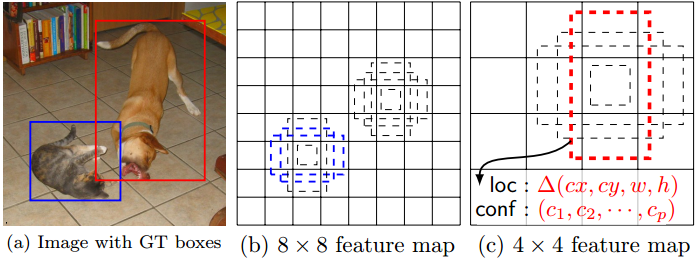
4、本文的这些改进设计，能够在当输入分辨率较低的图像时，保证检测的精度。同时这个整体端对端的设计，训练也变得简单。在检测速度、检测精度之间取得较好的权衡；

5、本文提出的模型在不同的数据集上，如PASCAL VOC、MS COCO、ILSVRC，都进行了测试。在检测时间、检测精度上，均与目前物体检测领域最好的检测方法进行了比较；

## 2、细节

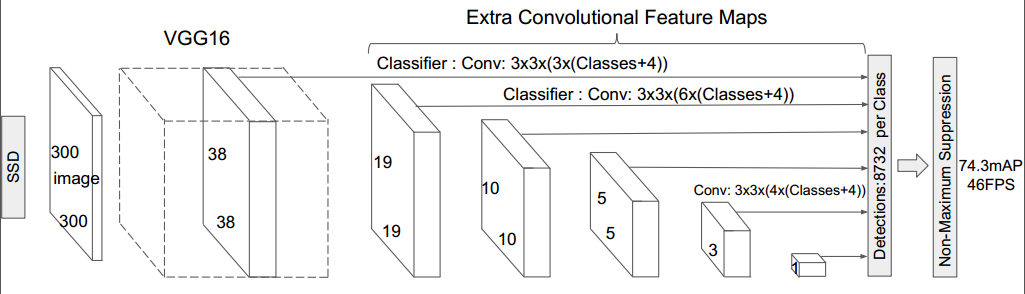
目前流行的优秀的检测系统大致都是如下步骤，先生成一些假设的目标框，然后在这些目标框中提取特征，之后再经过一个分类器，来判断里面是不是物体，是什么物体。这类流程是从Selective Search开始，到如今在PASCAL VOC、MS COCO、ILSVRC数据集上取得领先的是基于ResNet的Faster R-CNN。但这类方法对于嵌入式系统，所需要的计算时间太久了，不足以实时的进行检测。当然也有很多工作是朝着实时检测迈进，但目前为止，都是牺牲检测精度来换取时间。

本文提出的实时检测方法，消除了中间的目标框、特征重采样的过程。虽然本文不是第一篇这样做的文章，但本文做了一些提升性的工作，既保证了速度，也保证了检测精度。这部分详细讲解了SSD物体检测框架，以及SSD的训练方法。先弄清楚下文所说的默认目标框和特征图单元是什么。看下图：特征图单元就是将特征图切分成8×8或者4×4之后的一个个格子；默认目标框就是每一个格子上，一系列固定大小的框，即图中虚线所形成的一系列框；



SSD算法在训练阶段需要输入原图像和标准框信息，然后就可以在网络的最后输出每个default box属于某一类别的概率及其相对于默认框的偏移量。在训练时，首先会根据标准框去匹配默认框，从而确定哪些default box属于正样本，哪些属于负样本。

### 2.1 Model



SSD是基于一个前向传播卷积神经网络网络，产生一系列固定大小的目标框和每一个目标框中包含物体实例的可能性，即得分。之后，进行一个非极大值抑制得到最终的预测结果。SSD模型的最开始部分，本文称作基础网络，是用于图像分类的标准架构。在基础网络之后，本文添加了额外辅助的网络结构：

1、多尺度的特征图

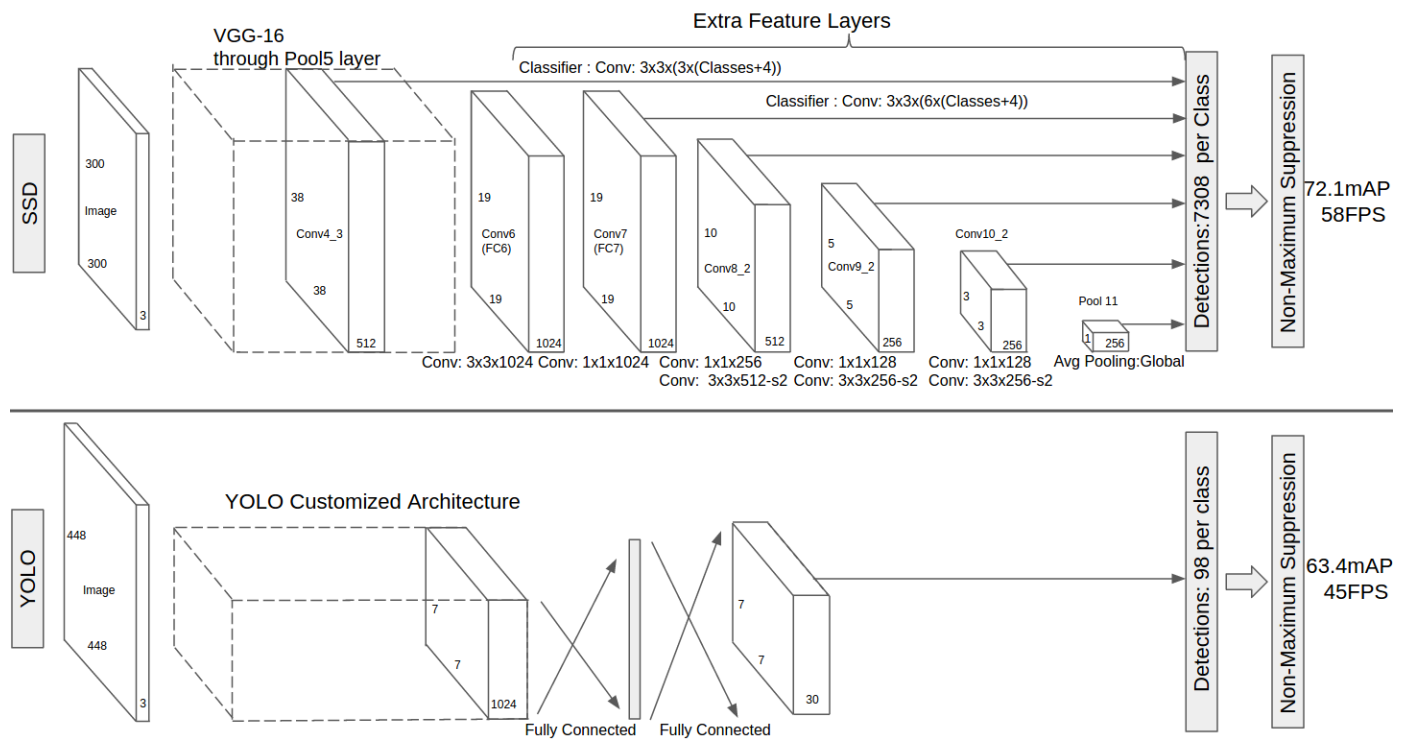
在基础网络结构后，添加了额外的卷积层，这些卷积特征层的大小是逐层递减的，可以在多尺度下进行目标预测，而Overfeat和YOLO都是在单尺度特征图上进行预测的。

2、多个卷积预测器

每一个添加的特征层或者在基础网络结构中的特征层，可以使用一系列卷积滤波器去产生一系列固定大小的预测，具体见上图的网络架构图。对于一个大小为，具有p通道的特征层，使用的卷积核滤波器就是的核。产生的预测，要么是归属类别的一个得分，要么是相对于默认框坐标的偏移量。在每一个的特征图位置上，使用上面的大小的核，就会产生一个输出值。目标框偏移值是输出的默认框与此时特征图之间的相对距离，YOLO架构则是用一个全连接层来代替这里的卷积层。

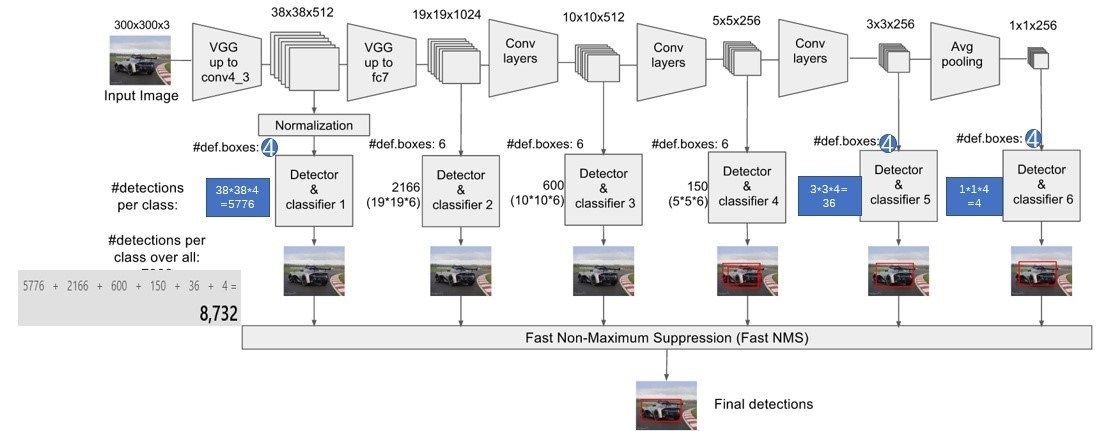
3、默认框和不同的纵横比

每一个box相对于与其对应的特征图单元的位置是固定的。在每一个特征图单元中，我们要预测得到的框与默认框之间的偏移量，以及每一个框中包含物体的得分即每一个类别概率都要计算出。因此，对于一个位置上的k个框中的每一个框，我们需要计算出c个类，每一个类的得分，还有这个框相对于它的默认框的4个偏移值。于是，在特征图中的每一个特征图单元上，就需要有个滤波器。对于一张大小的特征图，即会产生 个输出结果。这里的默认框类似于Faster R-CNN中的Anchor boxes，关于这里的Anchor boxes，详细的参见原论文。但是又不同于Faster R-CNN，本文中的Anchor boxes用在了不同分辨率的特征图上。



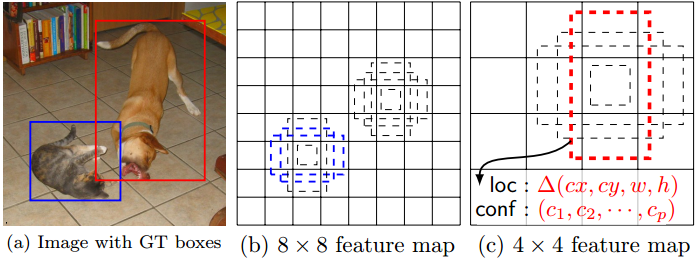
上图为SSD与YOLO网络框架的对比，从结果来看，SSD比YOLO检测效果更好，同时检测速度更快。SSD算法的关键点之一在于网络结构的优化。采用VGG16的基础网络结构，使用前面的5层，然后利用astrous算法将fc6 和fc7层转化成两个卷积层。再格外增加了3个卷积层，和一个平均池化层。不同层次的特征图分别用默认框的偏移以及不同类别得分的预测，通过非极大值抑制得到最终的检测结果。

这些增加的卷积层的特征图的大小变化比较大，允许能够检测出不同尺度下的物体： 在低层的特征图，感受野比较小，高层的感受野比较大，在不同的特征图进行卷积，可以达到多尺度的目的。观察YOLO，后面存在两个全连接层，全连接层以后，每一个输出都会观察到整幅图像，并不是很合理。但是SSD去掉了全连接层，每一个输出只会感受到目标周围的信息，包括上下文。这样来做就增加了合理性。并且不同的特征图，预测不同宽高比的图像，这样比YOLO增加了预测更多的比例的box。下图是横向的流程：



### 2.2 Train

在训练时，本文的SSD与那些用区域候选框加池化方法的区别是：SSD训练图像中的标准框需要赋予到那些固定输出的框上。在前面也已经提到了，SSD 输出的是事先定义好的，一系列固定大小的目标框。



如上图中，狗的标准框是红色的目标框，但在进行标注的时候，要将红色的标准框赋予图c中一系列固定输出的框中的一个，即图c中的红色虚线框。事实上，文章中指出，像这样定义的标准框不止在本文中用到。在 YOLO 、Faster R-CNN中的区域候选框阶段以及在MultiBox中都用到了。

当这种将训练图像中的标准框与固定输出的框对应之后，就可以端对端的进行损失函数的计算以及反向传播的计算更新了。训练中会遇到一些问题：1、选择一系列默认框；2、选择上文中提到的尺度的问题；3、难例挖掘；4、数据增广的策略，下面会谈本文的解决这些问题的方式，分为以下下面的几个部分：

1、标准框与默认框的匹配策略

将标准框与默认框进行配对，以组成训练标签，在开始的时候，用 MultiBox中的best jaccard overlap来匹配每一个标准框与默认框，这样就能保证每一个标准框与唯一的一个默认框对应起来。但是又不同于MultiBox，本文之后又将默认框与任何的标准框配对，只要两者之间的jaccard overlap大于一个阈值，这里本文的阈值为 0.5，就认为该默认框在训练时是正样本，否则是负样本，这样的设置可以简化学习问题。

2、训练目标

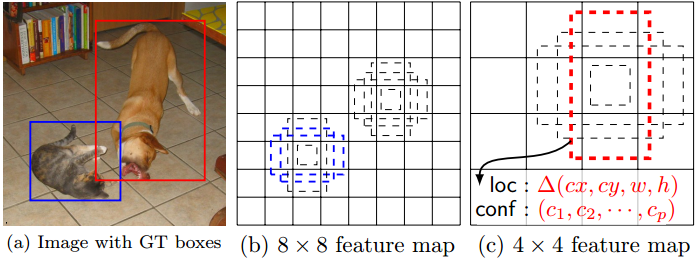
SSD训练的目标函数源自于MultiBox的目标函数，但是本文将其拓展，使其可以处理多个目标类别。用表示 第i个默认框与类别为p的第j个标准框相匹配，否则若不匹配的话，则。根据上面的匹配策略，一定有，意味着对于第j个标准框有可能有多个默认框与其相匹配。那么总的目标损失函数就由与的加权求和：

其中：N是与标准框相匹配的默认框个数。localization loss是 Fast R-CNN 中 Smooth L1 Loss，用在预测框与标准框参数即中心坐标位置，宽和高中，回归目标框的中心位置，以及宽度和高度。confidence loss是 Softmax Loss，输入为每一类的置信度c，权重项 α，设置为1。

3、选择默认框的尺度和纵横比

大部分卷积神经网络在越深的层，特征图的尺寸会越来越小。这样做不仅仅是为了减少计算与内存的需求，还有个好处就是最后提取的特征图就会有某种程度上的平移与尺度不变性。同时为了处理不同尺度的物体，一些文章将图像转换成不同的尺度，将这些图像独立的通过卷积网络处理，再将这些不同尺度的图像结果进行融合。如果使用同一个网络中的、不同层上的特征图，也可以达到相同的效果，同时在所有物体尺度中共享参数。之前的工作就用了卷积网络前面的层来提高图像分割的效果，因为越底层的层，保留的图像细节越多。文章也证明了以上的想法是可行的。

因此，本文同时使用底层特征图和高层特征图来做目标检测。下图展示了本文中使用的两种不同尺度的特征图，8×8的特征图，以及 4×4的特征图：



一般来说，一个卷积网络中不同的层有着不同尺寸的感受野。这里的感受野，指的是输出的特征图上的一个节点，其对应输入图像上尺寸的大小。所幸的是，SSD 结构中，默认框不必要与每一层的感受野对应。本文的设计中，特征图中特定的位置，来负责图像中特定的区域，以及物体特定的尺寸。加入我们用m个特征图来做预测，每一个特征图中默认框的尺寸大小计算如下：

取值0.2，取值 0.95，意味着最低层的尺度是0.2，最高层的尺度是0.95，不同纵横比的默认框用来表示，则每一个默认框的宽度和高度就可以计算出来。对于纵横比为1时，本文还增加了一个默认框。所以最终在每个特征图位置上，有6个默认框。

在结合所有特征图上所有的不同尺度、不同纵横比的默认框，它们预测结果之后。可以想见有许多个预测值，包含了物体的不同尺寸、形状。如图，狗的标准框与4×4特征图中的红色框吻合，所以其余的框都看作负样本。

4、难例挖掘

在生成一系列的预测之后，会产生很多个符合标准框的预测框。同时，不符合标准框的也很多，而且这个负样本框远多于正样本框。这会造成正负样本的不均衡使得训练时难以收敛。因此，本文采取先将每一个物体位置上对应预测框是负的框进行排序，按照默认框的置信度的大小选择最高的几个，保证最后正负样本框的比例在3:1。本文通过实验发现，这样的比例可以更快的优化，训练也更稳定。

5、数据增广

本文同时对训练数据做了数据增广。关于数据增广，推荐一篇文章：Must Know Tips/Tricks in Deep Neural Networks，其中的第一部分就讲了数据增广技术。每一张训练图像，随机的进行如下几种选择：

1、使用原始的图像；

2、采样一个图像块其最小的jaccard overlap为：0.1、0.3、0.5、0.7与 0.9；

3、随机的采样一个图像块；

采样的图像块是原始图像大小比例是[0.1，1]，纵横比在1/2与2之间。当标准框的中心在采样的图像块中时，我们保留重叠部分。在这些采样步骤之后，每一个采样的图像块被调整大小到固定的大小，并且以0.5的概率随机的水平翻转；

## 3、实验

实验是检验真理的唯一标准，那么对实验结果详细的分析以及结合算法的原理对算法本质上的一些思考有利于之后研究工作的开展，也是今后工作的一个研究突破点。

### 3.1 安装SSD

该部分主要参考SSD官方github上的相关说明。首先需要安装SSD的caffe框架：

git clone <https://github.com/weiliu89/caffe.git>

cd caffe

git checkout ssd

然后编译caffe框架：

cp Makefile.config.example Makefile.config

make -j4

make pycaffe

接下来准备SSD的预训练模型：

下载在目录caffe/models/VGGNet中得到基础网络：

VGG\_ILSVRC\_16\_layers\_fc\_reduced.caffemodel

然后按照训练需求下载VOC2007和VOC2012的数据集；

最后在训练之前准备LMDN文件，需要运行./data/VOC0712/create\_list.sh脚本用来在文件夹/voc0712/中生成trainval.txt和test.txt以及test\_name\_size.txt文件。最后再运行create\_data.sh脚本得到VOC0712\_trainval\_lmdb和VOC0712\_test\_lmdb

### 3.2 训练SSD

直接运行python脚本example/ssd/ssd\_pascal.py文件执行SSD的训练过程。可以执行脚本example/sssd/score\_ssd\_pascal.py可以评估训练结果。如果要测试SSD模型，可以运行脚本example/ssd/ssd\_pascal\_webcom.py在主机摄像头上测试算法。

### 3.3 训练自己的数据集

1、准备原始图片数据

原始数据的来源包括：网络爬虫、PASCAL VOC07-12、USV0414、VID2016、ImageNet得到所有原始数据后，都先预处理成500长度的大小。

2、标注数据集（VOC2007格式）

使用的工具是开源软件LabelImg可以得到各个图片的标注XML文件。

3、分割数据集

模型训练时，需要分割成训练集、测试集和验证集，这些操作只需要执行脚本文件create\_trainval.txt可以得到4个txt文件。

最后在训练之前准备LMDN文件，需要运行./data/VOC0712/create\_list.sh脚本用来在文件夹/voc0712/中生成trainval.txt和test.txt以及test\_name\_size.txt文件。最后再运行create\_data.sh脚本得到VOC0712\_trainval\_lmdb和VOC0712\_test\_lmdb

### 3.4 一些思考

当下流行的目标检测算法大体可以分为两类：

1、Region Proposal-based：这种方法首先要获得proposals，然后对proposals再进行分类，具有代表性的有R-CNN系列及其扩展。这种方法的优点的是检测准精度高，但是速度慢，无法达到实时性，速度几乎取决于基础网络的复杂度1FPS~17FPS不等，速度越快，精确度越低。

2、Proposal-free：这种方法在检测时不依赖于Proposals，在检测时同时对物体进行定位和识别，具有代表性的有YOLO系列、SSD及其扩展(DSSD、DSOD)等。这种方法的优点是检测速度快，可以做到实时检测，同时大目标的检测精度可以与第一类相媲美，但是小目标的检测效果较差。

无论是Region-based还是Proposal-free的检测方法，都比较依赖于其base network，如Alexnet、VGGnet、Resnet等等。换句话说，其检测的精度和速度都和基础网络的性能、复杂度有直接关联，网络性能越好，检测的精度也会越高。不同的检测方法可以通过替换base network的方式在速度和精度上进行权衡，而在比较时，也尽量分析相同base network的检测结果。SSD是一个非常经典的Proposal-free的检测方法，它直接以图片作为输入，同时输出物体的边界框及类别。SSD方法出现在YOLO V1之后，在保证实时检测的同时，对检测精度的精度进行了提升，在VOC、COCO数据集上可以与同时期最好的Region-based方法相媲美。

总结全文，在以下三个方面需要明确：

1、整体的网络结构，对VGG有哪些修改

2、检测流程，如何在feature map上进行卷积来实现default box的位置回归及分类

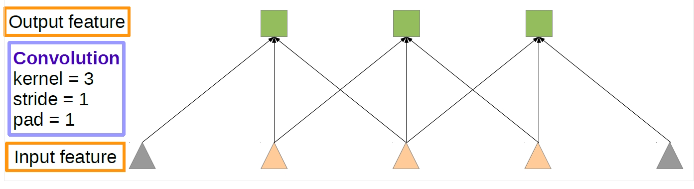
3、具体训练细节

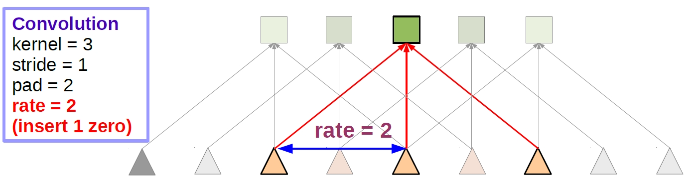
详细说明如下：

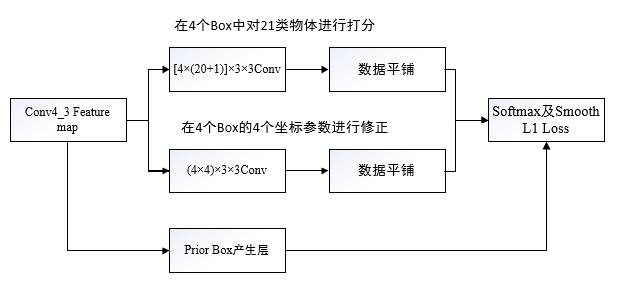
1、SSD的基础网络为VGG16，其作出的修改为将原网络第6、7层的4096维全连接层变成了3×3×1024和1×1×1024的卷积层，并去掉了第8个全连接层、dropout操作和最后的softamx层。

值得一提的是，作者还对Pool5（第5个池化层，在原fc6之前）进行了修改，将池化核大小从2×2变为3×3，步长由2变为1，padding由0变为1。这样做是为了保证feature map的尺度不会过于小，避免了过多的信息损失（特别是小目标的信息），但是同时会减小感受野。所以，作者还额外使用了atrous算法来避免感受野变小，在这里简单介绍下atrous算法。

Atrous算法也叫hole filling算法。如何增加感受野呢？就是将卷积核膨胀。普通的卷积核是在map上紧密相连的像素上加权，如下图。在该方法中，卷积核作用的像素在map上是相隔的，相隔的大小称为hole size，也叫rate，如图所示。







上图是每个特征图上进行目标类别预测和位置回归的结构图。