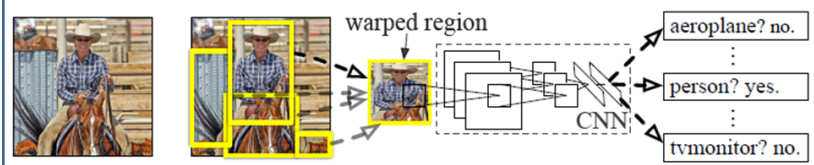


大话目标检测经典模型 (RCNN、Fast RCNN、Faster RCNN)

原创 雪饼 工作日志 2018/03/31 14:47 阅读数 15.1W

目标检测经典模型

RCNN、Fast RCNN、Faster RCNN

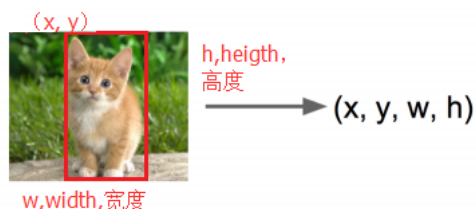


目标检测是深度学习的一个重要应用，就是在图片中要将里面的物体识别出来，并标出物体的位置，一般需要经过两个步骤：

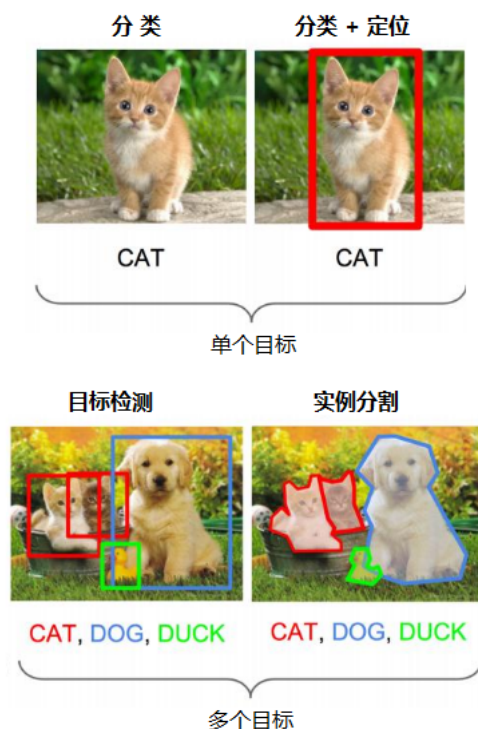
1、分类，识别物体是什么



2、定位，找出物体在哪里

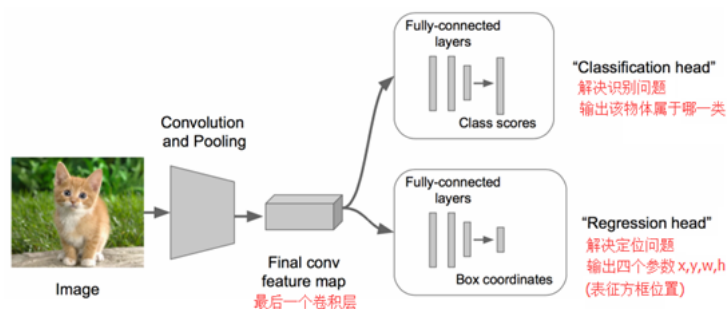


除了对单个物体进行检测，还要能支持对多个物体进行检测，如下图所示：



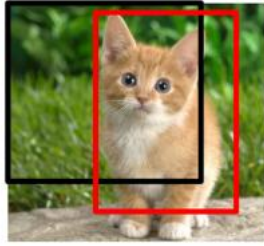
这个问题并不是那么容易解决，由于物体的尺寸变化范围很大、摆放角度多变、姿态不定，而且物体有很多种类别，可以在图片中出现多种物体、出现在任意位置。因此，目标检测是一个比较复杂的问题。

最直接的方法便是构建一个深度神经网络，将图像和标注位置作为样本输入，然后经过CNN网络，再通过一个分类头（Classification head）的全连接层识别是什么物体，通过一个回归头（Regression head）的全连接层回归计算位置，如下图所示：



但“回归”不好做，计算量太大、收敛时间太长，应该想办法转为“分类”，这时容易想到套框的思路，即取不同大小的“框”，让框出现在不同的位置，计算出这个框的得分，然后取得分最高的那个框作为预测结果，如下图所示：

左上角的黑框：得分0.5

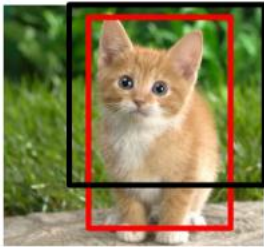


Larger image:
3 x 257 x 257

0.5	

Classification scores:
P(cat)

右上角的黑框：得分0.75

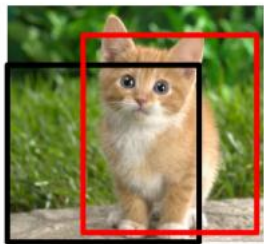


Larger image:
3 x 257 x 257

0.5	0.75

Classification scores:
P(cat)

左下角的黑框：得分0.6

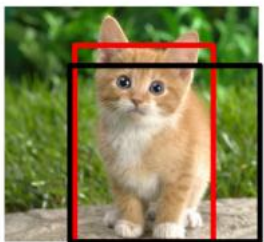


Larger image:
3 x 257 x 257

0.5	0.75
0.6	

Classification scores:
P(cat)

右下角的黑框：得分0.8



Larger image:
3 x 257 x 257

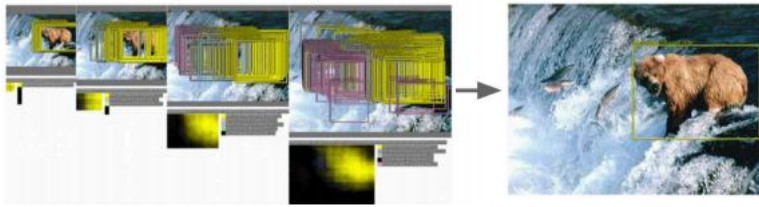
0.5	0.75
0.6	0.8

Classification scores:
P(cat)

根据上面比较出来的得分高低，选择了右下角的黑框作为目标位置的预测。

但问题是：框要取多大才合适？太小，物体识别不完整；太大，识别结果多了很多其它信息。那怎么办？那就各种大小的框都取来计算吧。

如下图所示（要识别一只熊），用各种大小的框在图片中进行反复截取，输入到CNN中识别计算得分，最终确定出目标类别和位置。

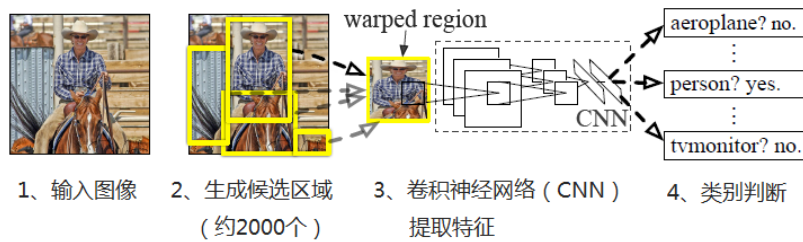


这种方法效率很低，实在太耗时了。那有没有高效的目标检测方法呢？

一、R-CNN 横空出世

R-CNN (Region CNN, 区域卷积神经网络) 可以说是利用深度学习进行目标检测的开山之作，作者Ross Girshick多次在PASCAL VOC的目标检测竞赛中折桂，2010年更是带领团队获得了终身成就奖，如今就职于Facebook的人工智能实验室 (FAIR) 。

R-CNN算法的流程如下



- 1、输入图像
- 2、每张图像生成1K~2K个候选区域
- 3、对每个候选区域，使用深度网络提取特征 (AlexNet、VGG等CNN都可以)
- 4、将特征送入每一类的SVM 分类器，判别是否属于该类
- 5、使用回归器精细修正候选框位置

下面展开进行介绍

1、生成候选区域

使用Selective Search (选择性搜索) 方法对一张图像生成约2000-3000个候选区域，基本思路如下：

- (1) 使用一种过分割手段，将图像分割成小区域
- (2) 查看现有小区域，合并可能性最高的两个区域，重复直到整张图像合并成一个区域位置。优先合并以下区域：

- 颜色 (颜色直方图) 相近的
- 纹理 (梯度直方图) 相近的
- 合并后总面积小的
- 合并后，总面积在其BBOX中所占比例大的

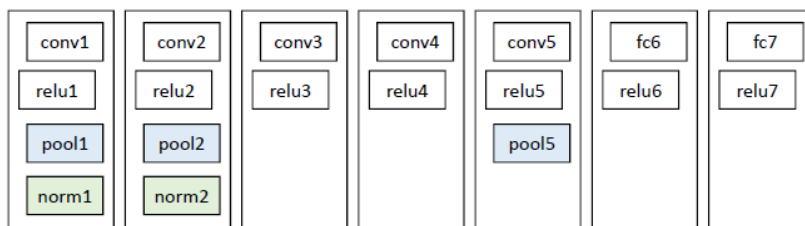
在合并时须保证合并操作的尺度较为均匀，避免一个大区域陆续“吃掉”其它小区域，保证合并后形状规则。

- (3) 输出所有曾经存在过的区域，即所谓候选区域

2、特征提取

使用深度网络提取特征之前，首先把候选区域归一化成同一尺寸227×227。

使用CNN模型进行训练，例如AlexNet，一般会略作简化，如下图：

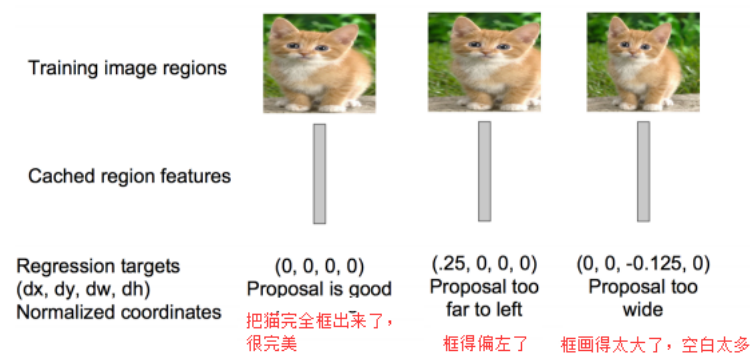


3、类别判断

对每一类目标，使用一个线性SVM二类分类器进行判别。输入为深度网络 (如上图的AlexNet) 输出的4096维特征，输出是否属于此类。

4、位置精修

目标检测的衡量标准是重叠面积：许多看似准确的检测结果，往往因为候选框不够准确，重叠面积很小，故需要一个位置精修步骤，对于每一个类，训练一个线性回归模型去判定这个框是否框得完美，如下图：



R-CNN将深度学习引入检测领域后，一举将PASCAL VOC上的检测率从35.1%提升到53.7%。

二、Fast R-CNN大幅提速

继2014年的R-CNN推出之后，Ross Girshick在2015年推出Fast R-CNN，构思精巧，流程更为紧凑，大幅提升了目标检测的速度。

Fast R-CNN和R-CNN相比，训练时间从84小时减少到9.5小时，测试时间从47秒减少到0.32秒，并且在PASCAL VOC 2007上测试的准确率相差无几，约在66%-67%之间。

		R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	9.5 hours
	(Speedup)	1x	8.8x
FASTER!	Test time per image	47 seconds	0.32 seconds
	(Speedup)	1x	146x

Fast R-CNN主要解决R-CNN的以下问题：

1、训练、测试时速度慢

R-CNN的一张图像内候选框之间存在大量重叠，提取特征操作冗余。而Fast R-CNN将整张图像归一化后直接送入深度网络，紧接着送入从这幅图像上提取出的候选区域。这些候选区域的前几层特征不需要再重复计算。

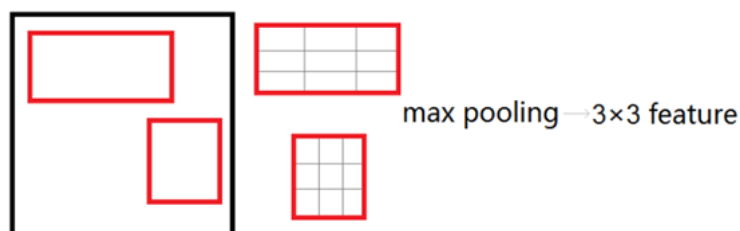
2、训练所需空间大

R-CNN中独立的分类器和回归器需要大量特征作为训练样本。Fast R-CNN把类别判断和位置精调统一用深度网络实现，不再需要额外存储。

下面进行详细介绍

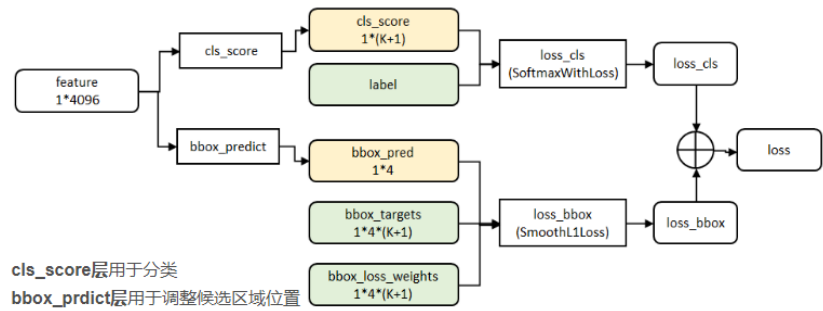
1、在特征提取阶段，通过CNN（如AlexNet）中的conv、pooling、relu等操作都不需要固定大小尺寸的输入，因此，在原始图片上执行这些操作后，输入图片尺寸不同将会导致得到的feature map（特征图）尺寸也不同，这样就不能直接接到一个全连接层进行分类。

在Fast R-CNN中，作者提出了一个叫做ROI Pooling的网络层，这个网络层可以把不同大小的输入映射到一个固定尺度的特征向量。ROI Pooling层将每个候选区域均匀分成M×N块，对每块进行max pooling。将特征图上大小不一的候选区域转变为大小统一的数据，送入下一层。这样虽然输入的图片尺寸不同，得到的feature map（特征图）尺寸也不同，但是可以加入这个神奇的ROI Pooling层，对每个region都提取一个固定维度的特征表示，就可再通过正常的softmax进行类型识别。



2、在分类回归阶段，在R-CNN中，先生成候选框，然后再通过CNN提取特征，之后再用SVM分类，最后再做回归得到具体位置（bbox regression）。而在Fast R-CNN中，作者巧妙的把最后的bbox regression也放进了神

经网络内部，与区域分类合并成为了一个multi-task模型，如下图所示：



实验表明，这两个任务能够共享卷积特征，并且相互促进。

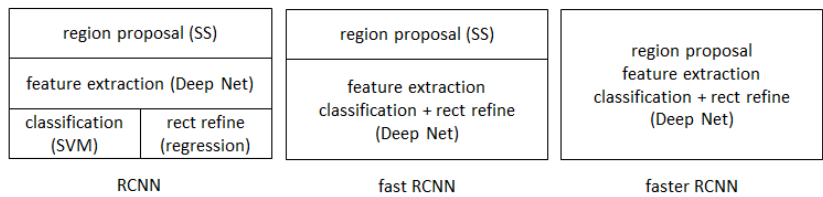
Fast R-CNN很重要的一个贡献是成功地让人们看到了Region Proposal+CNN（候选区域+卷积神经网络）这一框架实时检测的希望，原来多类检测真的可以在保证准确率的同时提升处理速度。

三、Faster R-CNN更快更强

继2014年推出R-CNN，2015年推出Fast R-CNN之后，目标检测界的领军人物Ross Girshick团队在2015年又推出一力作：Faster R-CNN，使简单网络目标检测速度达到17fps，在PASCAL VOC上准确率为59.9%，复杂网络达到5fps，准确率78.8%。

在Fast R-CNN还存在着瓶颈问题：Selective Search（选择性搜索）。要找出所有的候选框，这个也非常耗时。那我们有没有一个更加高效的方法来求出这些候选框呢？

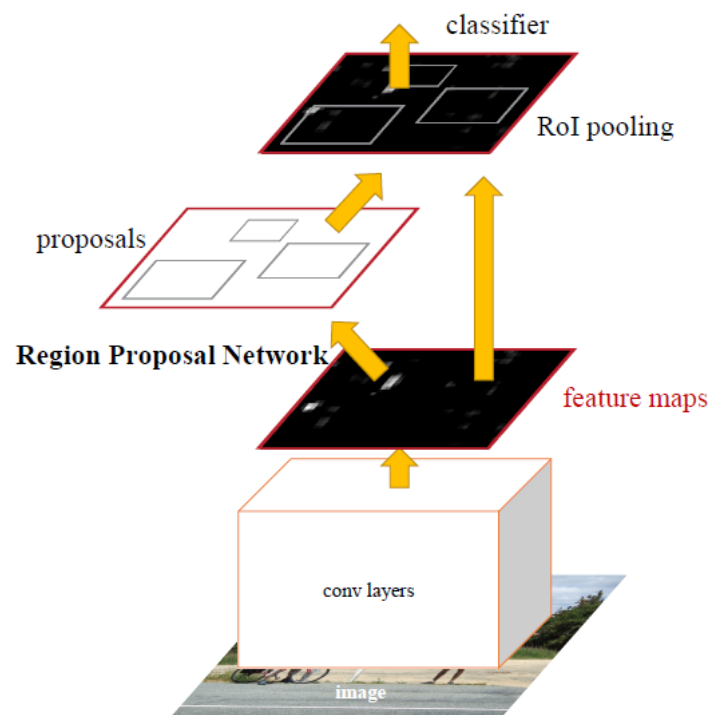
在Faster R-CNN中加入一个提取边缘的神经网络，也就说找候选框的工作也交给神经网络来做了。这样，目标检测的四个基本步骤（候选区域生成，特征提取，分类，位置精修）终于被统一到一个深度网络框架之内。如下图所示：



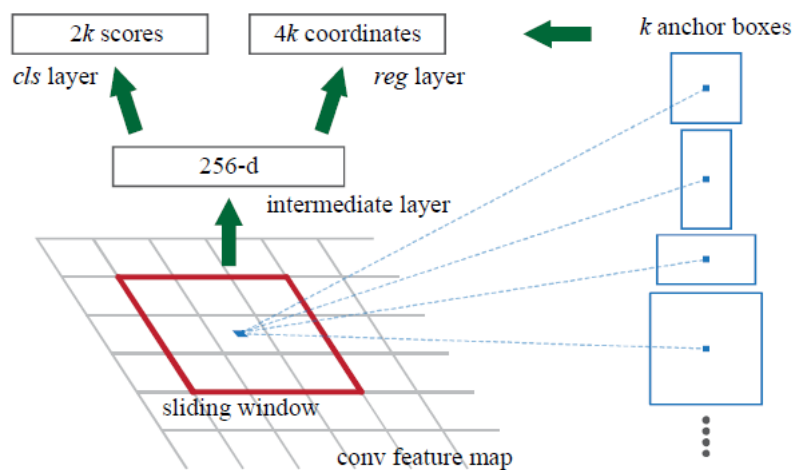
Faster R-CNN可以简单地看成是“区域生成网络+Fast R-CNN”的模型，用区域生成网络（Region Proposal Network，简称RPN）来代替Fast R-CNN中的Selective Search（选择性搜索）方法。

如下图

Faster R-CNN:



RPN如下图：



Region Proposal Network (RPN)

RPN的工作步骤如下：

- 在feature map（特征图）上滑动窗口
- 建一个神经网络用于物体分类+框位置的回归
- 滑动窗口的位置提供了物体的大体位置信息
- 框的回归提供了框更精确的位置

Faster R-CNN设计了提取候选区域的网络RPN，代替了费时的Selective Search（选择性搜索），使得检测速度大幅提升，下表对比了R-CNN、Fast R-CNN、Faster R-CNN的检测速度：

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	0.2 seconds
(Speedup)	1x	25x	250x
mAP (VOC 2007)	66.0	66.9	66.9

总结

R-CNN、Fast R-CNN、Faster R-CNN一路走来，基于深度学习目标检测的流程变得越来越精简、精度越来越高、速度也越来越快。基于region proposal（候选区域）的R-CNN系列目标检测方法是目标检测技术领域中的最主要分支之一。

墙裂建议

2014至2016年，Ross Girshick 等人发表了关于R-CNN、Fast R-CNN、Faster R-CNN的经典论文《Rich feature hierarchies for accurate object detection and semantic segmentation》、《Fast R-CNN》、《Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks》，在这些论文中对目标检测的思想、原理、测试情况进行了详细介绍，建议阅读这些篇论文以全面了解目标检测模型。

关注本人公众号“大数据与人工智能Lab”（BigdataAILab），然后回复“**论文**”关键字可在线阅读经典论文的内容。

推荐相关阅读

- [大话卷积神经网络（CNN）](#)

- [大话循环神经网络（RNN）](#)
- [大话深度残差网络（DRN）](#)
- [大话深度信念网络（DBN）](#)
- [大话CNN经典模型：LeNet](#)
- [大话CNN经典模型：AlexNet](#)
- [大话CNN经典模型：VGGNet](#)
- [大话CNN经典模型：GoogLeNet](#)
- [浅说“迁移学习”](#)
- [什么是“强化学习”](#)
- [AlphaGo算法原理浅析](#)
- [大数据究竟有多少个V](#)
- [Apache Hadoop 2.8 完全分布式集群搭建超详细教程](#)
- [Apache Hive 2.1.1 安装配置超详细教程](#)
- [Apache HBase 1.2.6 完全分布式集群搭建超详细教程](#)
- [离线安装Cloudera Manager 5和CDH5（最新版5.13.0）超详细教程](#)

[目标检测](#)

[深度学习](#)

[RCNN](#)

[Fast RCNN](#)

[Faster RCNN](#)

作者的其它热门文章

[大话CNN经典模型：GoogLeNet（从Inception v1到v4的演进）](#)

[大话卷积神经网络（CNN）](#)

[大话CNN经典模型：VGGNet](#)

[大话深度残差网络（DRN）ResNet网络原理](#)