对象检测

曹一 21721162

摘要

对象检测一直是计算机视觉领域的重要研究课题。要让计算机识别和理解图片或视频中的物体，这需要我们能够定位目标物体，并且对其进行分类。定位需要我们在物体周围画出一个边界框。随着深度学习的发展，它也在对象检测领域表现出了杰出的效果，现行的很多高效而又精准的方法都是使用了深度学习技术。在这篇文章中，我将会介绍近年来在对象检测领域比较成功的几个方法。

1.导论

在计算机视觉领域，让机器理解图片内容一直是很多研究者所希望做到的。人类可以快速轻松地识别出图片中的物体内容与信息，但是对于计算机要去理解像素之间的结构关系和像素集合产生的整体的信息是非常困难的。

近几年，深度学习技术开始在各个领域表现出极大的优势，对象检测领域也随着卷积神经网络的流行开始突飞猛进。尤其是在 2014 年所提出的 R-CNN[1]相比以往的算法实现了极大的提升，它结合了选择性搜索的区域提名（selective search region proposals）和卷积神经网络。从此之后一些列的基于神经网络的算法开始如雨后春笋般呈现出来，在效率和准确率上都节节拔升。

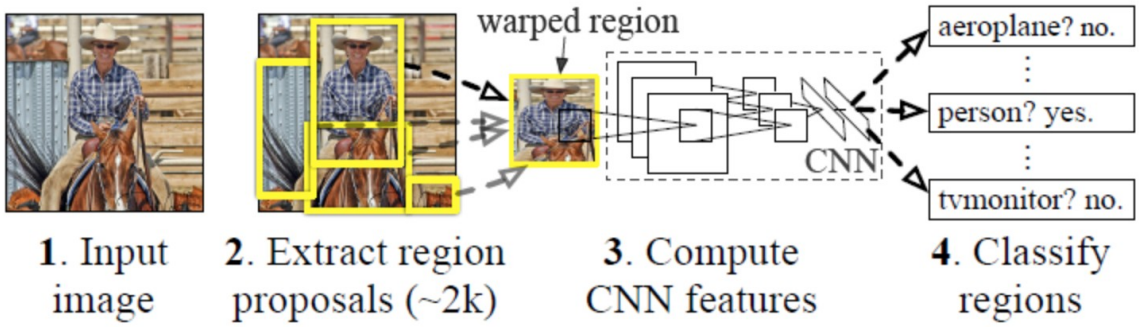
Fast R-CNN[2] 在单个的过程中使用多任务的损失函数来训练神经网络，这个网络结构是基于原有的 R-CNN。Faster R-CNN[3]用更快的神经网络技术来取代了缓慢的选择性搜索的区域提名（selective search region proposals）。R_FCN[4]相比较 Faster R-CNN 有着更快的处理速度。YOLO(You Only Look Once) [5]是在 2016 年 CVPR 上提出的一个完全不同的方法，将对象检测作为一个回归问题。SSD[6]结合了 Faster R-CNN 和 YOLO 的核心思路，集两者之长处。

2.方法

对象检测算法的进步是分为两个阶段的，一个是基于传统的特征，一个是基于深度学习。基于传统的优化算法在 2013 年以前还是主流。但是在 2013 年之后，整个学术界还有工业界逐渐开始使用深度学习技术，因为相比较传统的对象检测方法，深度学习技术能够更加高效准确的实现目标。

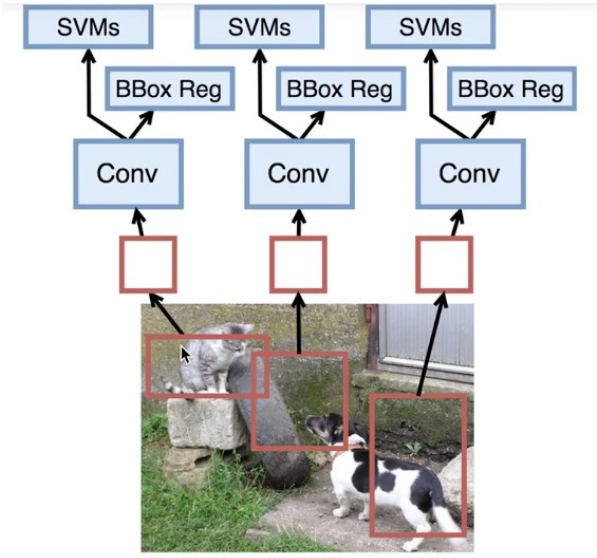
2.1.R-CNN

R-CNN 是在 2014 年 CVPR 上由 Ross Girshick 提出的方法。这个对象检测系统包括了三个模块。第一个模块是选择性搜索的区域提名（selective search region proposals），这些区域是检测器的候选目标。第二个模块是一个大型的卷积神经网络，它可以从每个候选区域中挖掘出一个定长的特征向量。第三个模块式线性的支持向量机（Support Vector Machine ，SVM），这三个模块的作用可以被归纳为三个步骤。首先是推荐区域的产生：使用选择性搜索方法（selective search）算法扫描输入的图片产生 2000 个推荐区域。然后特征挖掘：在这些推荐区域上运行卷积神经网络。最后对象检测和定位：将卷积神经网络的输出结果喂给支持向量机进行分类。



图一：R-CNN 过程示意图

在第一步中，每个区域的大小是不同的，但是卷积神经网络的输入要求固定大小的图片。所以我们必须要将不同尺寸的图片切片转化成为固定的大小，可以采取扭曲或者切割的办法实现。第二个模块中的卷积神经网络结构式 Alexnet，为了能够让这个网络适应新的任务和领域，论文中作者使用随机梯度下降算法继续训练网络的参数。最终 R-CNN 在 VOC 2012 数据集上实现了 53.3%的 mAP，这比当时其他最好的算法提升了 30%。使用 GPU 处理速度是每张图片 13 秒，使用 CPU 处理速度是每张图片 53 秒。和之前的工作比较起来，R-CNN 有着更加杰出的对象检测精确度。

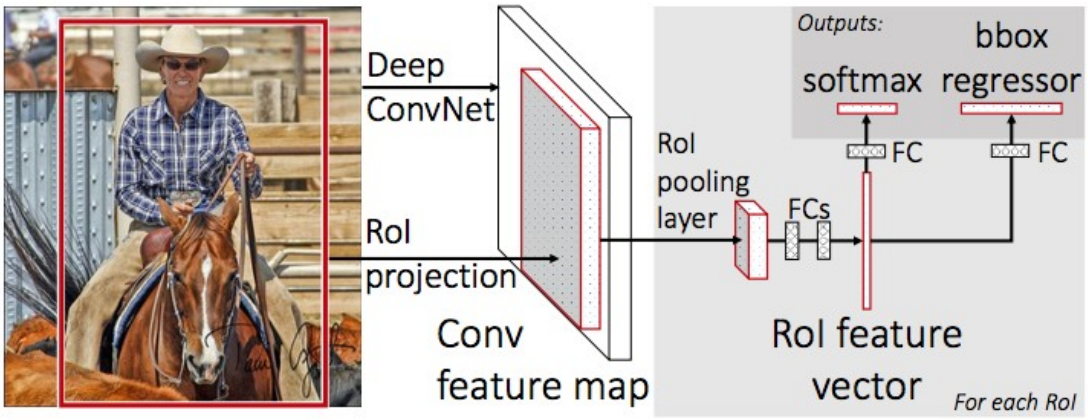


图二： R-CNN 的一个实例

但是它也有着很明显的缺点。第一，训练是一个多层次的过程，训练卷积神经网络再有支持向量机，最后边界框回归。第二，训练过程耗费时间与空间。由于这是个对层次的步骤，所以我们需要将中间的结果保存起来，存在磁盘上。这样的处理操作就会使用的大量的空间，并且也会耗费大量时间。第三，测试检测速度缓慢。除了前面的训练耗时耗费空间以外，这个算法在测试过程中同样非常缓慢。因为在测试过程中，每张图片会形成大量的推荐区域，而其又独立地处理这些推荐区域，这个过程中有大量的重复计算部分。当然，这些问题都在之后的工作中得到了改进。

2.2.Fast R-CNN

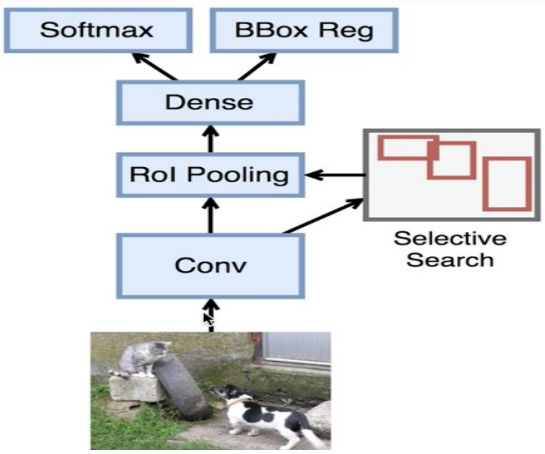
Fast R-CNN 是在 R-CNN 基础之上发展得到的新方法，比起以往它获得了更快的处理效率。这主要是有两个方面的原因：首先，在产生推荐区域之前就对图片进行了特征提取，因此不用重复独立的计算 2000 个推荐区域，而只用对每张图片进行一次卷积网络的特征提取操作。然后，将支持向量机用 softmax 层替换了，扩展了神经网络的功能。



图三：Fast R-CNN 过程示意图

Fast R-CNN 网络将整个图像和一组候选框作为输入。网络首先使用几个卷积层（conv）和最大池化层来处理整个图像，以产生卷积特征图。然后，对于每个候选框，RoI 池化层从特征图中提取固定长度的特征向量。每个特征向量被送入一系列全连接（fc）层中，其最终分支成两个同级输出层 ：一个输出 KK 个类别加上 1 个背景类别的 Softmax 概率估计，另一个为 KK 个类别的每一个类别输出四个实数值。每组 4 个值表示 KK 个类别的一个类别的检测框位置的修正。

RoI 池化层使用最大池化将任何有效的 RoI 内的特征转换成具有 H×WH×W（例如，7×77×7）的固定空间范围的小特征图，其中 HH 和 WW 是层的超参数，独立于任何特定的 RoI。



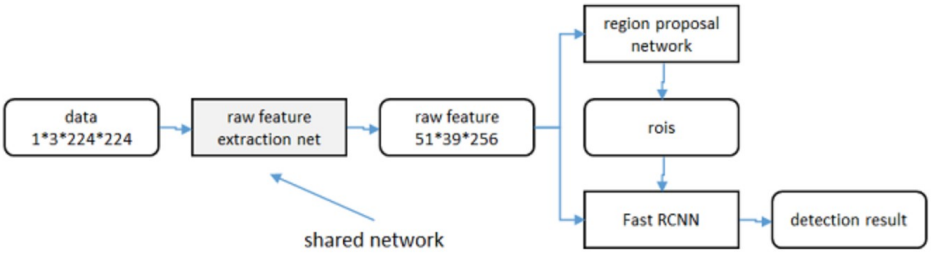
图四：Fast R-CNN 一个实例

正如我们所看到的，我们是在挖掘出的特征之上产生推荐区域。另外，使用 softmax 替代支持向量机，我们就只有一个神经网络需要去训练了。Fast R-CNN 在 VOC 2012 数据集

上实现了 66%mAP。处理速度是每张图片 0.3 秒。

2.3.Faster R-CNN

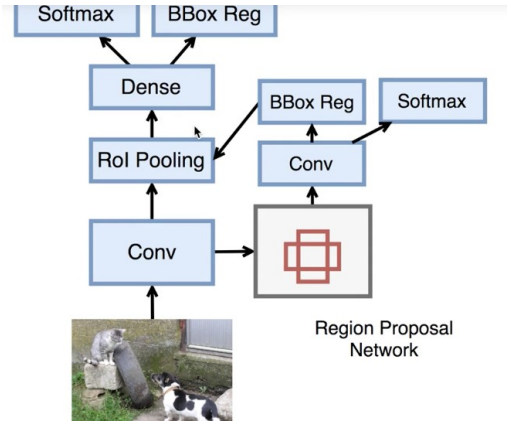
目前最先进的目标检测网络需要先用区域建议算法推测目标位置，像 SPPnet[6]和 Fast R-CNN[1]这些网络已经减少了检测网络的运行时间，这时计算区域建议就成了瓶颈问题。本文中，我们介绍一种区域建议网络（Region Proposal Network, RPN），它和检测网络共享全图的卷积特征，使得区域建议几乎不花时间。RPN 是一个全卷积网络，在每个位置同时预测目标边界和 objectness 得分。RPN 是端到端训练的，生成高质量区域建议框，用于 Fast R-CNN 来检测。通过一种简单的交替运行优化方法，RPN 和 Fast R-CNN 可以在训练时共享卷积特征。对于非常深的 VGG-16 模型，我们的检测系统在 GPU 上的帧率为 5fps（包含所有步骤），在 PASCAL VOC 2007 和 PASCAL VOC 2012 上实现了最高的目标检测准确率（2007 是 73.2%mAP，2012 是 70.4%mAP），每个图像用了 300 个建议框。代码已公开。



图五：Faster R-CNN 网络结构

区域建议网络（RPN）将一个图像（任意大小）作为输入，输出矩形目标建议框的集合，每个框有一个 objectness 得分。我们用全卷积网络对这个过程构建模型，本章会详细描述。因为我们的最终目标是和 Fast R-CNN 目标检测网络共享计算，所以假设这两个网络共享一系列卷积层。

为了生成区域建议框，我们在最后一个共享的卷积层输出的卷积特征映射上滑动小网络，这个网络全连接到输入卷积特征映射的 nxn 的空间窗口上。每个滑动窗口映射到一个低维向量上（对于 ZF 是 256-d，对于 VGG 是 512-d，每个特征映射的一个滑动窗口对应一个数值）。这个向量输出给两个同级的全连接的层——包围盒回归层（reg）和包围盒分类层（cls）。本文中 n=3，注意图像的有效感受野很大（ZF 是 171 像素，VGG 是 228 像素）。图 1（左）以这个小网络在某个位置的情况举了个例子。注意，由于小网络是滑动窗口的形式，所以全连接的层（nxn 的）被所有空间位置共享（指所有位置用来计算内积的 nxn 的层参数相同）。这种结构实现为 nxn 的卷积层，后接两个同级的 1x1 的卷积层（分别对应 reg 和 cls），ReLU 应用于 nxn 卷积层的输出。

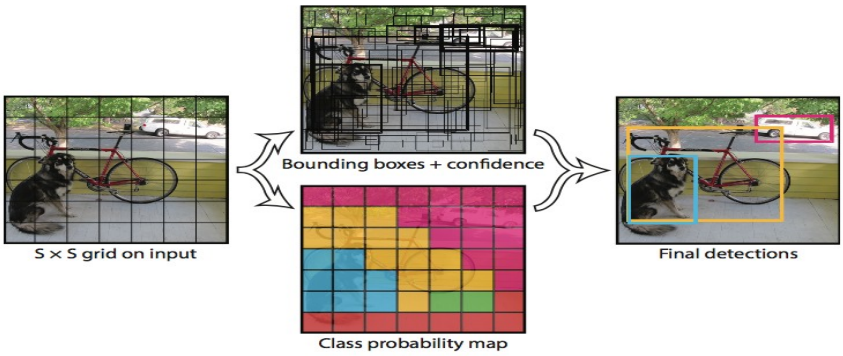


图六：Faster R-CNN 一个实例

最后 Faster R-CNN 在 VOC 2007 上实现了 73.2%，在 VOC 2012 上实现了 70.4%的 mAP。速度能够达到每秒 5 帧。在值得一提的是，虽然之后还是有很多的工作来提出更快的检测速度，但是 Faster R-CNN 仍然是有着非常好的性能的算法之一。

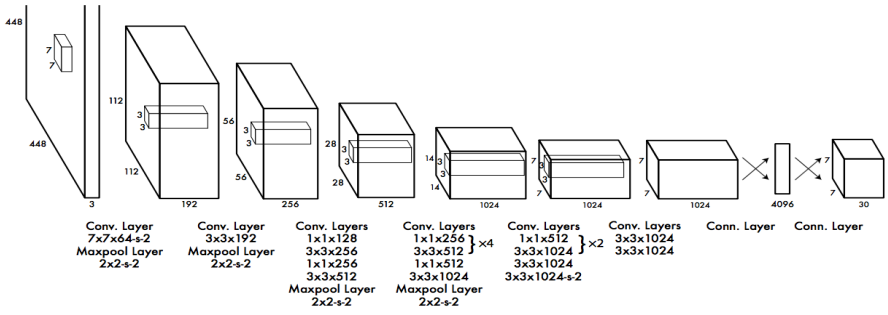
2.4.YOLO

YOLO（you only look once），一种新的目标检测方法。以前的目标检测工作重新利用分类器来执行检测。相反，将目标检测框架看作回归问题从空间上分割边界框和相关的类别概率。单个神经网络在一次评估中直接从完整图像上预测边界框和类别概率。由于整个检测流水线是单一网络，因此可以直接对检测性能进行端到端的优化。这个统一架构非常快。YOLO 模型以 45 帧/秒的速度实时处理图像。网络的一个较小版本，快速 YOLO，每秒能处理惊人的 155 帧，同时实现其它实时检测器两倍的 mAP。与最先进的检测系统相比，YOLO 产生了更多的定位误差，但不太可能在背景上的预测假阳性。最后，YOLO 学习目标非常通用的表示。当从自然图像到艺术品等其它领域泛化时，它都优于其它检测方法，包括 DPM 和 R-CNN。



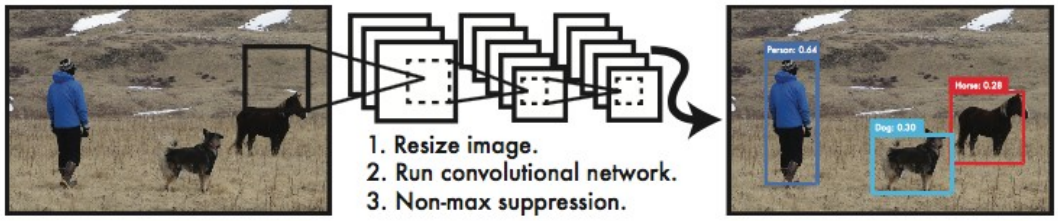
图七：YOLO 示意图

YOLO 将目标检测的单独组件集成到单个神经网络中。使用整个图像的特征来预测每个边界框。它还可以同时预测一张图像中的所有类别的所有边界框。这意味着我们的网络全面地推理整张图像和图像中的所有目标。YOLO 设计可实现端到端训练和实时的速度，同时保持较高的平均精度。将输入图像分成 S×S 的网格。如果一个目标的中心落入一个网格单元中，该网格单元负责检测该目标。每个网格单元预测这些盒子的 B 个边界框和置信度分数。这些置信度分数反映了该模型对盒子是否包含目标的信心，以及它预测盒子的准确程度。在形式上，我们将置信度定义为 Pr(Object) * IOU_predtruth。如果该单元格中不存在目标，则置信度分数应为零。否则，我们希望置信度分数等于预测框与真实值之间联合部分的交集（IOU）。



图八：YOLO 网络结构

每个边界框包含 5 个预测：x，y，w，h 和置信度。(x，y)坐标表示边界框相对于网格单元边界框的中心。宽度和高度是相对于整张图像预测的。最后，置信度预测表示预测框与实际边界框之间的 IOU。每个网格单元还预测 C 个条件类别概率 Pr(Class_i|Object)。这些概率以包含目标的网格单元为条件。每个网格单元我们只预测的一组类别概率，而不管边界框的的数量 B 是多少。在测试时，乘以条件类概率和单个盒子的置信度预测，



图九：YOLO 一个实例

它为我们提供了每个框特定类别的置信度分数。这些分数编码了该类出现在框中的概率以及预测框拟合目标的程度。YOLO 最终在 VOC 2007 上实现了 63.4%的 mAP，YOLO 算法可以实现实时处理图片，达到每秒 45 帧。

3. 总结

这篇综述简要的介绍了 R-CNN, Fast R-CNN , Faster R-CNN, 和 YOLO。这些算法都是对象检测中非常流行和重要的模型，还有例如 R-FCN，SSD 这些模型也在这个方向上实现了很好的效果，也非常推荐读者进一步去了解。

参考文献

【1】Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.

【2】Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.、

【3】Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99,2015.

【4】Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. Rfcn: Object detection via region-based fully convolutional networks. In Advances in neural information processing systems, pages 379–387, 2016.

【5】Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, realtime object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 779–788, 2016.

【6】Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In European Conference on Computer Vision, pages 346–361. Springer, 2014.