

目标检测算法综述

Sigai 特约作者：东尼大佬

2018.10.12



原创声明：本文为 SIGAI 原创文章，仅供个人学习使用，未经允许，不得转载，不能用于商业目的。

导言

目标检测是计算机视觉和数字图像处理的一个热门方向，广泛应用于机器人导航、智能视频监控、工业检测、航空航天等诸多领域，通过计算机视觉减少对人力资本的消耗，具有重要的现实意义。因此，目标检测也就成为了近年来理论和应用的研究热点，它是图像处理和计算机视觉学科的重要分支，也是智能监控系统的核心部分，同时目标检测也是泛身份识别领域的一个基础性的算法，对后续的人脸识别、步态识别、人群计数、实例分割等任务起着至关重要的作用。由于深度学习的广泛运用，目标检测算法得到了较为快速的发展，本文广泛调研国内外目标检测方法，主要介绍基于深度学习的两种目标检测算法思路，分别为 One-Stage 目标检测算法和 Two-Stage 目标检测算法。

正文

1. 相关介绍

1.1 背景介绍

从 2006 年以来,在 Hilton、Bengio、LeChun 等人的引领下,大量深度神经网络的论文被发表,尤其是 2012 年, Hinton 课题组首次参加 ImageNet 图像识别比赛,其通过构建的 CNN 网络 AlexNet^[1]一举夺得冠军,从此神经网络开始受到广泛的关注。深度学习利用多层计算模型来学习抽象的数据表示,能够发现大数据中的复杂结构,目前,这项技术已成功地应用在包括计算机视觉领域在内的多种模式分类问题上。计算机视觉对于目标运动的分析可以大致分为三个层次:运动分割,目标检测;目标跟踪;动作识别,行为描述^[2]。其中,目标检测既是计算机视觉领域要解决的基础任务之一,同时它也是视频监控技术的基本任务。由于视频中的目标具有不同姿态且经常出现遮挡、其运动具有不规则性,同时考虑到监控视频的景深、分辨率、天气、光照等条件和场景的多样性,而且目标检测算法的结果将直接影响后续的跟踪、动作识别和行为描述的效果。故即使在技术发展的今天,目标检测这一基本任务仍然是非常具有挑战性的课题,存在很大的提升潜力和空间。

1.2 什么是目标检测

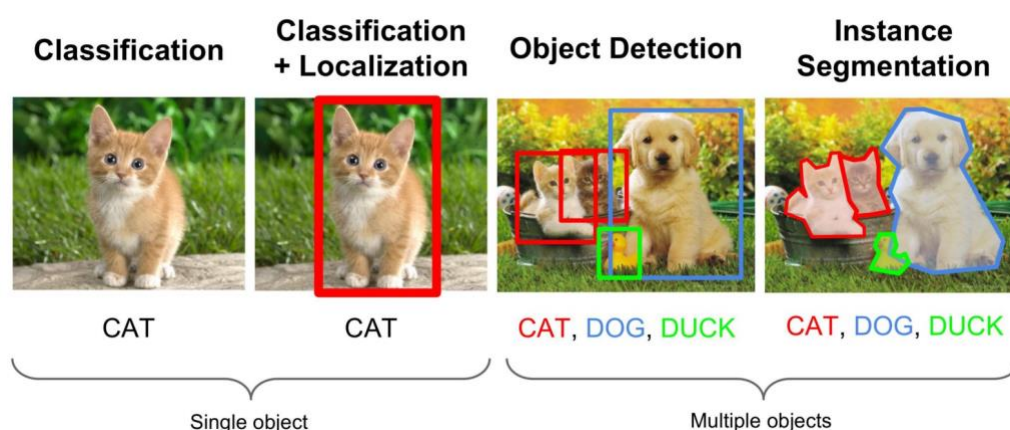


图 1: 任务对比图

目标检测即找出图像中所有感兴趣的物体,包含物体定位和物体分类两个子任务,同时确定物体的类别和位置。

1.3 传统算法概述

传统目标检测的方法一般分为三个阶段:首先在给定的图像上选择一些候选的区域,然后对这些区域提取特征,最后使用训练的分类器进行分类。下面我们对这三个阶段分别进行介绍。

(1) 区域选择:这一步是为了对目标的位置进行定位。由于目标可能出现在图像的任何位置,而且目标的大小、长宽比例也不确定,所以最初采用滑动窗口的策略对整幅图像进

行遍历,而且需要设置不同的尺度,不同的长宽比。这种穷举的策略虽然包含了目标所有可能出现的位置,但是缺点也是显而易见的:时间复杂度太高,产生冗余窗口太多,这也严重影响后续特征提取和分类的速度和性能。

(2) 特征提取:由于目标的形态多样性,光照变化多样性,背景多样性等因素使得设计一个鲁棒的特征并不是那么容易。然而提取特征的好坏直接影响到分类的准确性。其中,这个阶段常用的特征有 SIFT^[3]、HOG^[4]等。

(3) 分类:根据第二步提取到的特征对目标进行分类,分类器主要有 SVM,AdaBoost 等。

2. 基于深度学习的目标检测算法

2.1 算法概述

目标检测任务可分为两个关键的子任务:目标分类和目标定位。目标分类任务负责判断输入图像或图像区域中是否有感兴趣类别的物体出现,输出一系列带分数的标签表明感兴趣类别的物体出现在输入图像或图像区域中的可能性。目标定位任务负责确定输入图像或图像区域中感兴趣类别的物体的位置和范围,输出物体的包围盒、或物体中心、或物体的闭合边界等,通常方形包围盒是最常用的选择。

目前主流的目标检测算法主要是基于深度学习模型,其可以分成两大类:(1) One-Stage 检测算法,其不需要 Region Proposal 阶段,直接产生物体的类别概率和位置坐标值,比较典型的算法如 YOLO、SSD 和 CornerNet;(2) Two-Stage 目标检测算法,其将检测问题划分为两个阶段,首先产生候选区域(region proposals),然后对候选区域进行分类和位置精修,这类算法的典型代表是基于 region proposal 的 R-CNN 系列算法,如 R-CNN, Fast R-CNN, Faster R-CNN 等。目标检测模型的主要性能指标是检测准确度和速度,对于准确度,目标检测要考虑物体的定位以及分类准确度。一般情况下,Two-Stage 算法在准确度上有优势,而 One-Stage 算法在速度上有优势。不过,随着研究的发展,两类算法都在两个方面做改进,均能在准确度以及速度上取得较好的结果。

2.2 One-Stage 目标检测算法

One-Stage 目标检测算法可以在一个 stage 直接产生物体的类别概率和位置坐标值,相比于 Two-Stage 的目标检测算法不需要 Region Proposal 阶段,整体流程较为简单。如下图所示,在 Testing 的时候输入图片产生输出,解码生成对应检测框即可;在 Training 的

时候需要将 Ground Truth 编码成 CNN 输出对应的格式以便计算对应损失 loss。

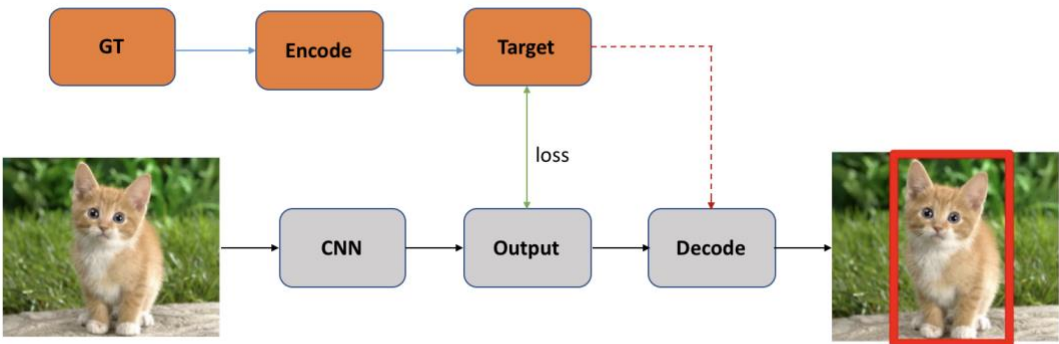


图 2：One-Stage 检测算法示意图

目前对于 One-Stage 算法的主要创新主要集中在如何设计 CNN 结构、如何构建网络目标以及如何设计损失函数上，接下来我将从这几个方面进行阐述。

2.2.1 如何设计 CNN 结构

设计 CNN 网络结构主要有两个方向，分别为追求精度和追求速度。最简单的一种实现方式就是替换 Backbone 网络结构，ResNet101 的表征能力要强于 MobileNet，然而 MobileNet 的计算量要远远低于 ResNet101，如果将 ResNet101 替换为 MobileNet，那么检测网络在精度应该会有一些的损失，但是在速度上会有一定提升；如果将 MobileNet 替换为 ResNet101，那么检测网络在速度上会有一定的损失，但是在精度上会有一些的提升。当然这只是一相对简单的改进 CNN 网络结构的方式，实际上在改进 CNN 结构的时候需要很多的学术积累和经验，我将通过几篇 SSD 相关论文做一下简要分析。

SSD^[5]：SSD 检测算法的网络结构如下图所示，其中 Backbone 为 VGG 网络，使用不同阶段不同分辨率的 feature map 进行预测。

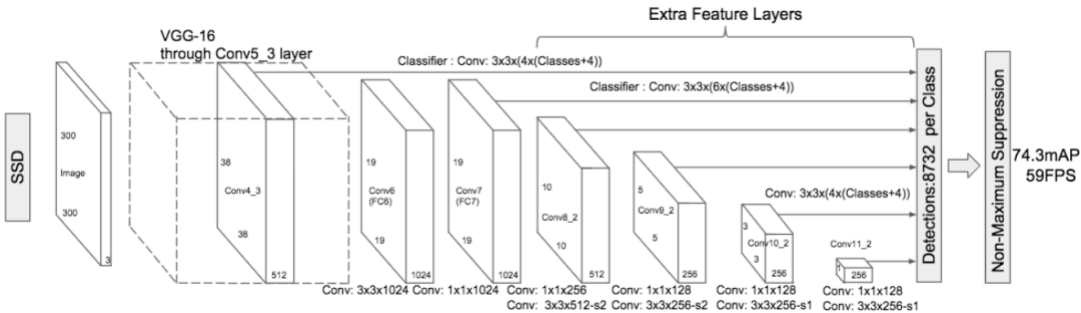


图 3：SSD 网络结构示意图

DSSD^[6]：DSSD 检测算法的网络结构如下图所示，DSSD 也是使用不同阶段不同分辨率的 feature map 进行预测，在不考虑 Backbone 网络结构差别的情况下，可以发现 DSSD 相比于 SSD 多了一系列的反卷积层，SSD 是使用下采样过程中的 feature map 进行预测，而 DSSD 是使用上采样过程中的 feature map 进行预测。SSD 用于检测的 feature map 位于网络的较低层，表征能力较弱，而 DSSD 用于检测的 feature map 位于网络的较高层，表征能力较强，同时 DSSD 在反卷积的过程中通过 Skip-Connection 引入了较低层的 feature map，实现了一定程度的特征融合。所以 DSSD 的效果要优于 SSD 检测算法。

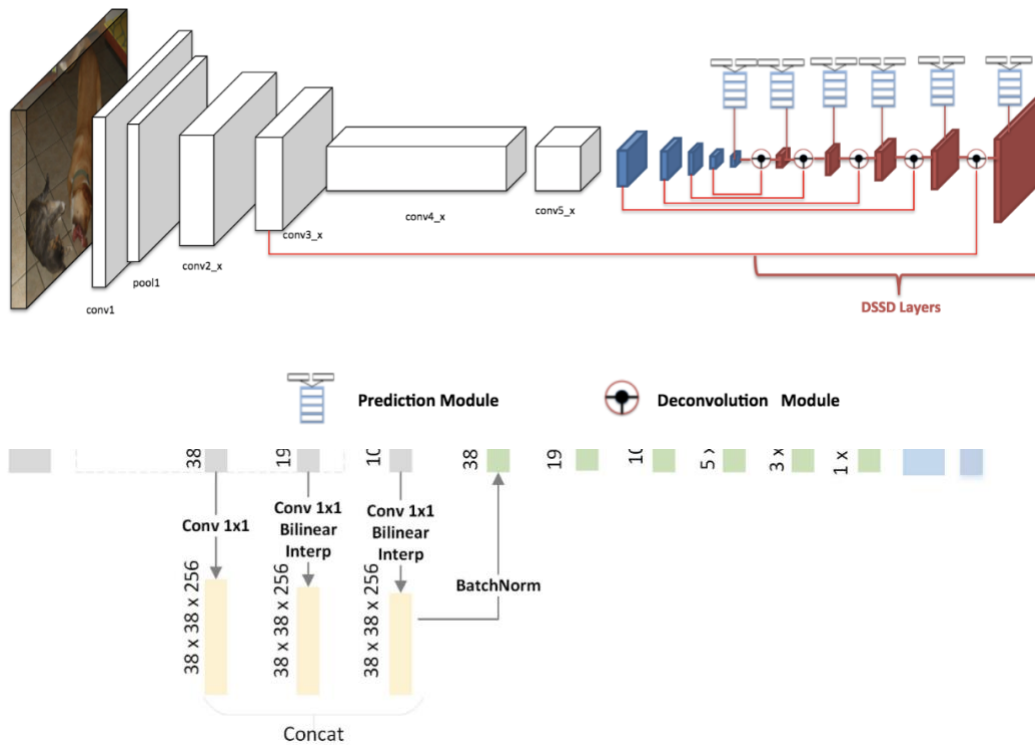


图 5：FSSD 网络结构示意图

2.2.2 如何构建回归目标

如何构建网络回归目标即如何区分正负样本使其与网络输出相对应，最简单直接的方法是直接回归物体的相关信息，除此之外，在回归坐标时可以回归物体坐标相对于 anchor 的偏移量等等。对于 One-Stage 检测方法主要有如下三种典型的回归目标构建方式，代表方法分别为 YOLO 系列算法、SSD 系列算法以及 CornerNet 目标检测算法。

YOLO 系列算法：如下图所示，其中左图取自 YOLOv1^[8]，右图取自 YOLOv2^[9]，需要说明的是 YOLOv1 相比于 YOLOv2 在坐标回归的时候没有 anchor 的概念。YOLO 系列算法在构建回归目标时一个主要的区别就是如果将图像划分成 $S \times S$ 的格子，每个格子只负责目标中心点落入该格子的物体的检测。

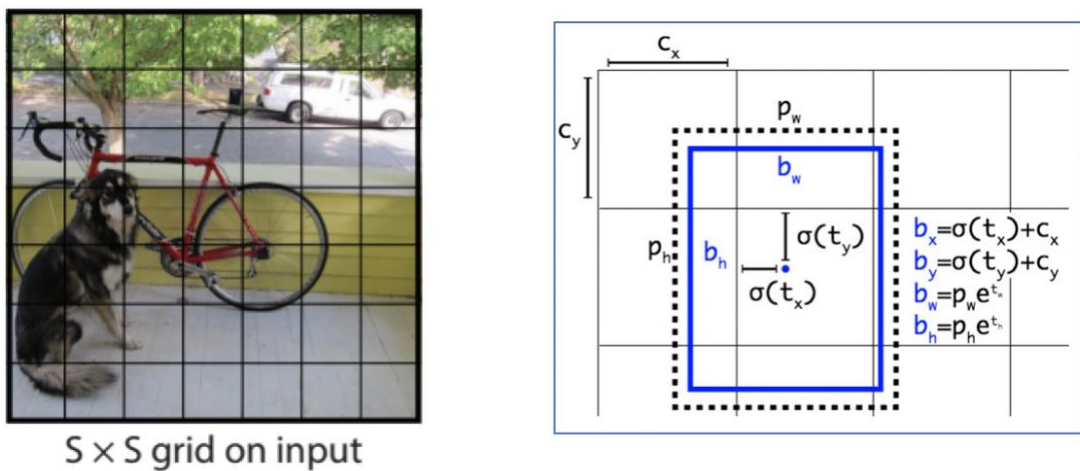


图 6：YOLO 系列算法正负样本区分图

SSD 系列算法：如下图所示，SSD 系列检测算法在确定正负样本的时候通过交并比大小进行区分，当某一个 Ground Truth 的目标框与 anchor 的交并比最大且对应的交并比大于某一个阈值的时候，对应 anchor 即负责检测该 Ground Truth。

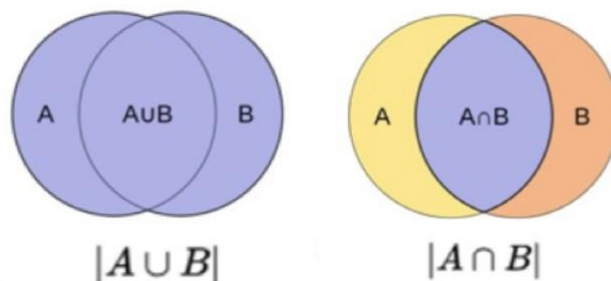


图 7：IOU（交并比）示意图

CornerNet^[10]：如下图所示，CornerNet 检测算法巧妙的将检测框转换成了关键点，即一个目标框可以由两个点来表示，那么在预测的时候就可以直接预测两个类别的关键点，然后对关键点进行组合即可。



图 8：CornerNet 目标示意图

2.2.3 如何设计损失函数

目标检测算法主要分为两个子任务，分别为物体分类和物体定位。损失主要包括分类损失和定位损失，常见的损失组合主要有如下两种 Cls Loss + Loc Loss（SSD 系列算法）、Cls Loss + Obj Loss + Loc Loss（YOLO 系列算法），即 YOLO 系列算法相比于 SSD 系列算法多了 Object Loss，即判断对应区域是否为物体的损失。除此之外，One-Stage 目标检测算法的正负样本不均衡的问题比较严重，对于设计损失函数还有一些针对创新。

Hard Negative Mining：即对于大量的负样本只挑取其中适当比例的损失较大的负样本计算损失，其余损失较小的负样本忽略不计；

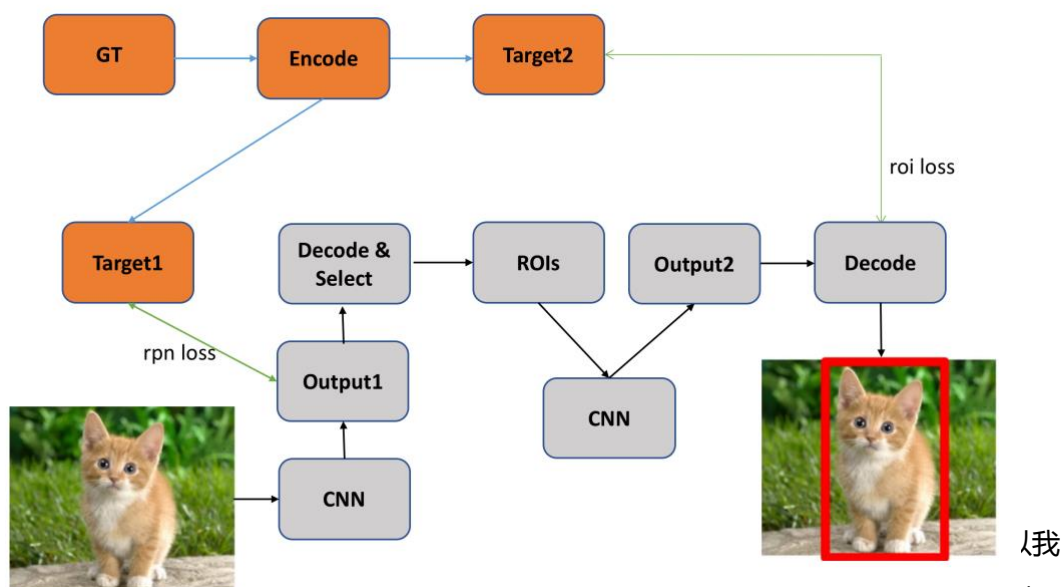
Focal Loss：由于大多数都是简单易分的负样本（属于背景的样本），使得训练过程不能充分学习到属于那些有类别样本的信息；其次简单易分的负样本太多，可能掩盖了其他有

类别样本的作用。Focal Loss 希望那些 hard examples 对损失的贡献变大，使网络更倾向于从这些样本上学习。

需要说明的是，个人任务 One-Stage 检测算法和 Two-Stage 检测算法的第一个 Stage 并没有太大区别，在某种程度上 Two-Stage 检测算法的第一个 Stage 可以看成是 One-Stage 检测算法，而第二个 Stage 只是对前一个 Stage 的结果做进一步精化，上述所有思路本人觉得都适用于 Two-Stage 检测算法的第一个 Stage。除此之外针对于 Two-Stage 检测框架设计的相关损失函数同样适用于 One-Stage 检测算法，如针对遮挡问题设计的相关 loss，具体不再阐述。

2.3 Two-Stage 目标检测算法

Two-Stage 目标检测算法可以看作是进行两次 One-Stage 检测，整体流程如下图所示，在 Testing 的时候输入图片产生 ROIs，然后对 ROIs 进一步精化产生最终输出，解码生成对应检测框即可；在 Training 的时候需要将 Ground Truth 编码成 CNN 输出对应的格式以便计算对应损失 loss。



我觉得 Two-Stage 可以由两个 One-Stage 检测算法的组合，第一个 Stage 做初步检测，剔除负样本，生成初步位置信息（Region of Interest），第二个 Stage 再做进一步精化并生成最终检测结果。目前对于 Two-Stage 算法的主要创新主要集中在如何高效准确地生成 Proposals、如何获取更好的 ROI features、如何加速 Two-Stage 检测算法以及如何改进后处理方法，接下来我将从这几个方面进行阐述。

2.3.1 如何高效准确地生成 Proposals

如何高效准确地生成 Proposals 考虑的是 Two-Stage 检测算法的第一个 Stage，获取初步的检测结果，供下一个 Stage 做进一步精化。接下来我将通过对比 R-CNN、Faster R-CNN 和 FPN 来做简要说明。

R-CNN^[11]：R-CNN 生成 Proposals 的方法是传统方法 Selective Search，主要思路是通过图像中的纹理、边缘、颜色等信息对图像进行自底向上的分割，然后对分割区域进行不同尺度的合并，每个生成的区域即一个候选 Proposal，如下图所示。这种方法速度较慢。

Bottom-up segmentation, merging regions at multiple scales

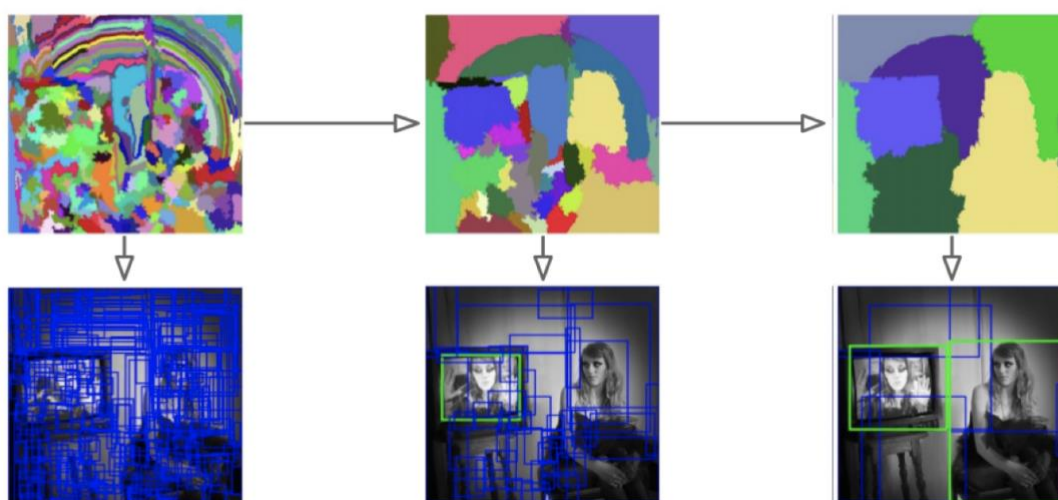


图 10：Selective Search 示意图

Faster R-CNN^[12]：Faster R-CNN 使用 RPN 网络代替了 Selective Search 方法，大大提高了生成 Proposals 的速度，具体实现策略同 One-Stage 检测算法，这里不再做过多赘述。网络示意图如下图所示。

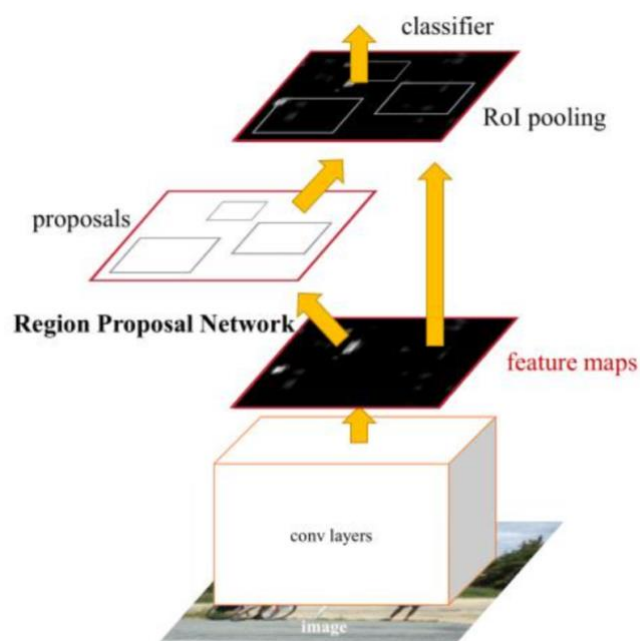


图 11：Faster R-CNN 网络示意图

FPN^[13]：Faster R-CNN 只采用顶层特征做预测，但我们知道低层的特征语义信息比较少，但是目标位置准确；高层的特征语义信息比较丰富，但是目标位置比较粗略。FPN 算法把低分辨率、高语义信息的高层特征和高分辨率、低语义信息的低层特征进行自上而下的侧边连接，使得所有尺度下的特征都有丰富的语义信息，然后在不同尺度的特征层上进行预测，使得生成 Proposals 的效果优于只在顶层进行预测的 Faster R-CNN 算法。如下图所示。

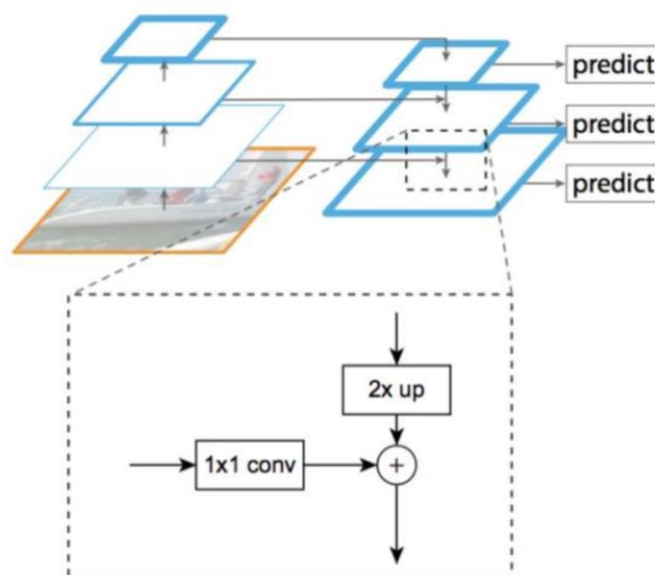


图 12：FPN Proposal 网络示意图

Cascade R-CNN^[14]：类似于 Faster R-CNN、FPN 等，其 Proposal 网络对于正样本只设置了一个阈值，只做了一次较为宽松的约束，得到的 Proposals 结果较为粗糙，当对检测框的定位结果要求更为精确的时候就稍显不足。而 Cascade R-CNN 在获取 Proposals 的时候也采用逐步求精的策略，前一步生成的 Proposals 作为后一步的输入，通过控制正样本的交并比阈值不断提高 Proposals 的质量，如下图所示。准确来说，Cascade R-CNN 应该不能算 Two-Stage 检测算法，应该是多 Stage 检测算法，多步求精。

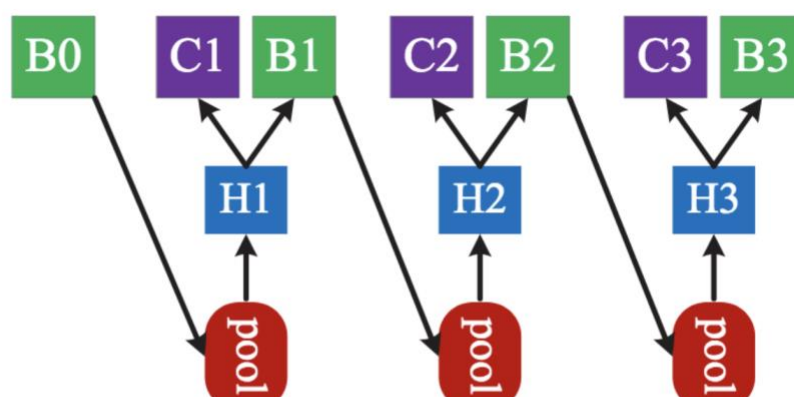


图 13 : Cascade R-CNN Proposal 网络示意图

2.3.2 如何获取更好的 ROI features

在获取 Proposals 之后，如何获取更好的 ROI features 是 Two-Stage 检测算法第二个 Stage 的关键，只有输入比较鲁棒的情况下才能得到较好的输出。对于这个问题主要考虑的有两个方向，其一是如何获取 Proposals 的 features，其二是得到 Proposals 的 features 之后如何 align 到同一个尺度。首先对于第一个问题主要有如下几种策略：

R-CNN：在原图裁剪 Proposals 对应区域，然后 align 到同一个尺度，分别通过神经网络提取特征；

Fast/Faster R-CNN：对原图提取一次特征，将对应 Proposals 的坐标映射到提取的 feature map 上提取对应特征；

FPN 对原图提取不同尺度的特征 将不同尺度的 Proposals 映射到不同尺度的 feature map 上提取对应特征；

R-FCN^[15]：同样对原图提取一次特征，主要区别是特征是 Position-Sensitive 的，即不同区域的特征维护在不同 channels 上，对于一个 Proposal，其不同区域区域的特征需要映射到不同的 channels 上，在提取目标特征的同时加上了位置信息。如下图所示。

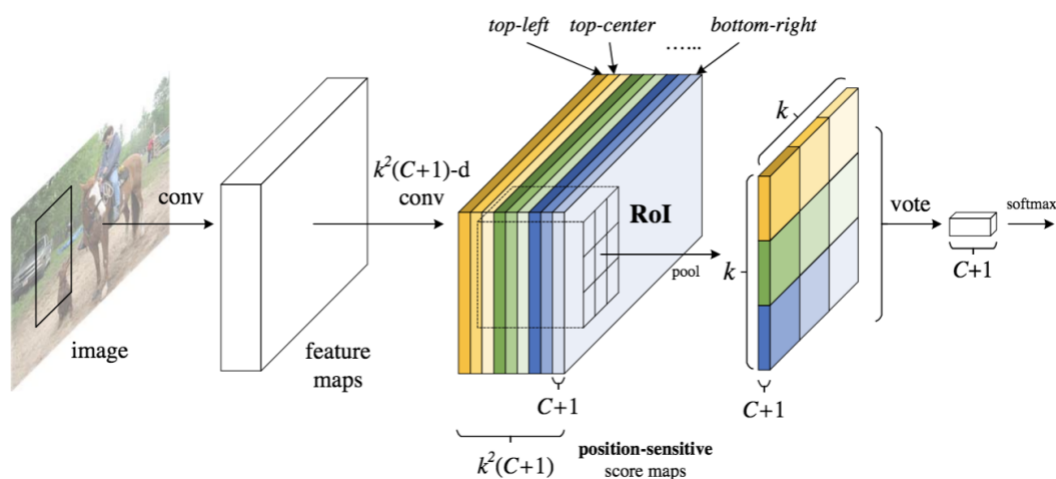


图 14：R-FCN 网络示意图

其次对于第二个问题主要有如下几种策略 分别为 ROI Pool、ROI Align、PSROI Pool、PrROI Pool，接下来做简要说明。

ROI Pool :即对 Proposal 区域分割固定大小的格子 ,每个格子做 Adaptive 的 Pooling 操作，获取一个值，从而所有 Proposal 生成同样大小的输出。

ROI Align：主要解决了 ROI Pool 的量化误差问题，即浮点数坐标转换成整数坐标产生的误差，主要解决方式即不采用量化方式获取具体坐标，每个位置的值采用双线性插值的方式获得。

PSROI Pool：即 R-FCN 采用的 Pooling 方式，与 ROI Pool 的唯一区别在于 PSROI Pool 需要每一个 Proposal 的不同区域对应到 feature map 的不同 channels 进行取值。

PrROI Pool：即 Precise ROI Pooling，其考虑了 Proposal 对应格子区域的每个值，采用积分的方式进行求解，而 ROI Align 只 Sample 对应格子区域的几个值。

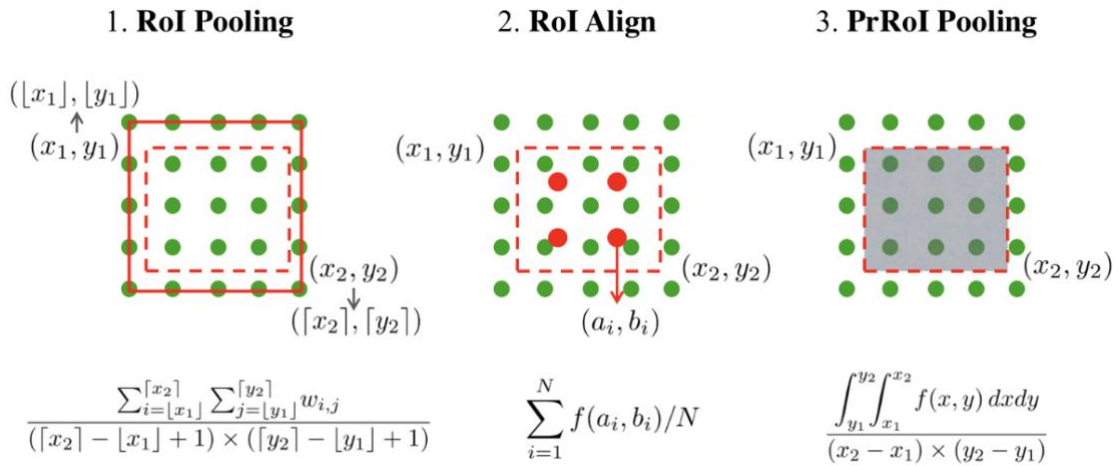


图 15：ROI 策略对比图

2.3.3 如何加速 Two-Stage 检测算法

Two-Stage 检测算法在一般情况下要慢于 One-Stage 检测算法，然而随着研究的发展，速度上的差别也在逐渐缩小，Two-Stage 算法的开销主要有两部分，一个是 Proposal 的开销，一个是 ROI Sub-Network 的开销，提高 Proposal 的效率和提高 ROI Sub-Network 的开销均可以加速 Two-Stage 检测算法。

Fast R-CNN vs. Faster R-CNN :Faster R-CNN 使用神经网络代替 Selective Search，大大提高了 Proposal 的速度，前面已有描述。

Faster R-CNN vs. Light-Head R-CNN^[16] : Light-Head R-CNN 使用更小的 Sub-

Network 代替 Faster R-CNN 较为臃肿的 Sub-Network，大大提高了 Two-Stage 检测算法的速度。

2.3.4 如何改进后处理方法

这里所说的后处理方法仅指 NMS 相关算法，NMS 即非极大值抑制算法，在目标检测，定位等领域是一种被广泛使用的方法，主要目的是为了消除多余的框，找到最佳的物体检测的位置，几乎所有的目标检测方法都用到了这种后处理算法。接下来我将简要介绍如下几种 NMS 相关算法，分别为 NMS、Soft NMS、Softer NMS、IOU-Guided NMS。

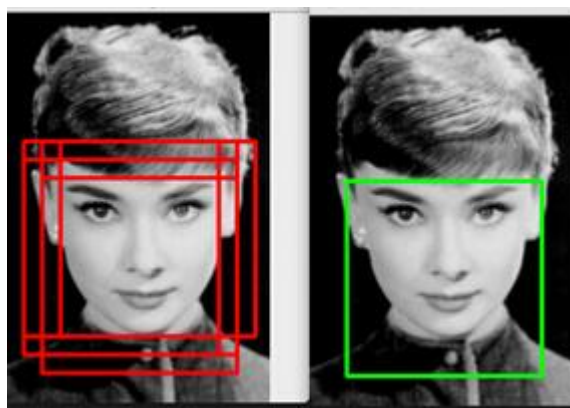


图 16：NMS 示例

Soft NMS^[17]：Soft NMS 相对于 NMS 的改进即每次并不是直接排除掉和已选框重叠大于一定阈值的框，而是以一定的策略降低对应框的 score，知道低于某个阈值，从而不至于过多删除拥挤情况下正确的框。

Softer NMS^[18]：Softer NMS 相对于 NMS 的改进即每次并不是直接以得分最大的框的坐标作为当前选择框的坐标，而是和得分最大的框重叠大于一定阈值的框的坐标进行一定策略的加权平均，所得的框作为当前选择的得分最大的框的坐标，从而尽可能地保留了最精确的检测框。

IOU-Guided NMS^[19]：即以 IOU 得分作为 NMS 的排序依据，因为 IOU 得分直接反应了对应框的定位精确程度，优先考虑定位精度较高的框，防止定位精度较低但是 Object 得分较高的框被误排序。

小结

目标检测至今仍然是计算机视觉领域较为活跃的一个研究方向，虽然 One-Stage 检测算法和 Two-Stage 检测算法都取得了很好的效果，但是对于真实场景下的应用还存在一定差距，目标检测这一基本任务仍然是非常具有挑战性的课题，存在很大的提升潜力和空间。

欢迎关注 <https://github.com/youansheng>

参考文献

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.
- [2] Vishwakarma S, Agrawal A. A survey on activity recognition and behavior understanding in video surveillance [J]. The Visual Computer, 2012: 1-27.
- [3] D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 2004.
- [4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.
- [5] Liu, Wei, et al. "SSD: Single Shot MultiBox Detector." European Conference on Computer Vision Springer International Publishing, 2016:21-37.
- [6] Fu C Y, Liu W, Ranga A, et al. DSSD : Deconvolutional Single Shot Detector[J]. 2017.
- [7] Li Z, Zhou F. FSSD: Feature Fusion Single Shot Multibox Detector[J]. 2017.
- [8] Redmon, Joseph, et al. "You Only Look Once: Unified, Real-Time Object Detection." (2015):779-788.
- [9] Redmon, Joseph, and A. Farhadi. "YOLO9000: Better, Faster, Stronger." (2016):6517-6525.
- [10] Law H, Deng J. CornerNet: Detecting Objects as Paired Keypoints[J]. 2018.
- [11] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR 2014.
- [12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Neural Information Processing Systems (NIPS), 2015.
- [13] Lin T Y, Dollár P, Girshick R, et al. Feature Pyramid Networks for Object Detection[J]. 2016.
- [14] Cai Z, Vasconcelos N. Cascade R-CNN: Delving into High Quality Object Detection[J]. 2017.
- [15] Dai J, Li Y, He K, et al. R-FCN: Object Detection via Region-based Fully Convolutional Networks[J]. 2016.

[16] Li Z, Peng C, Yu G, et al. Light-Head R-CNN: In Defense of Two-Stage Object Detector[J]. 2017.

[17] Bodla N, Singh B, Chellappa R, et al. Soft-NMS — Improving Object Detection with One Line of Code[J]. 2017.

[18] Softer-NMS: Rethinking Bounding Box Regression for Accurate Object Detection. arxiv id:1809.08545

[19] Jiang B, Luo R, Mao J, et al. Acquisition of Localization Confidence for Accurate Object Detection[J]. 2018. 推荐阅读

[1] [机器学习-波澜壮阔 40 年](#) 【获取码】SIGAI0413.

[2] [学好机器学习需要哪些数学知识?](#) 【获取码】SIGAI0417.

[3] [人脸识别算法演化史](#) 【获取码】SIGAI0420.

[4] [基于深度学习的目标检测算法综述](#) 【获取码】SIGAI0424.

[5] [卷积神经网络为什么能够称霸计算机视觉领域?](#) 【获取码】SIGAI0426.

[6] [用一张图理解 SVM 的脉络](#) 【获取码】SIGAI0428.

[7] [人脸检测算法综述](#) 【获取码】SIGAI0503.

[8] [理解神经网络的激活函数](#) 【获取码】SIGAI2018.5.5.

[9] [深度卷积神经网络演化历史及结构改进脉络-40 页长文全面解读](#) 【获取码】SIGAI0508.

[10] [理解梯度下降法](#) 【获取码】SIGAI0511.

[11] [循环神经网络综述—语音识别与自然语言处理的利器](#) 【获取码】SIGAI0515

[12] [理解凸优化](#) 【获取码】SIGAI0518

[13] [【实验】理解 SVM 的核函数和参数](#) 【获取码】SIGAI0522

[14] [【SIGAI 综述】行人检测算法](#) 【获取码】SIGAI0525

[15] [机器学习在自动驾驶中的应用—以百度阿波罗平台为例\(上\)](#) 【获取码】SIGAI0529

[16] [理解牛顿法](#) 【获取码】SIGAI0531

[17] [【群话题精华】5 月集锦—机器学习和深度学习中一些值得思考的问题](#) 【获取码】SIGAI 0601

[18] [大话 Adaboost 算法](#) 【获取码】SIGAI0602

[19] [FlowNet 到 FlowNet2.0: 基于卷积神经网络的光流预测算法](#) 【获取码】SIGAI0604

- [20] [理解主成分分析\(PCA\)](#) 【获取码】SIGAI0606
- [21] [人体骨骼关键点检测综述](#) 【获取码】SIGAI0608
- [22] [理解决策树](#) 【获取码】SIGAI0611
- [23] [用一句话总结常用的机器学习算法](#) 【获取码】SIGAI0611
- [24] [目标检测算法之 YOLO](#) 【获取码】SIGAI0615
- [25] [理解过拟合](#) 【获取码】SIGAI0618
- [26] [理解计算：从 \$\sqrt{2}\$ 到 AlphaGo ——第 1 季 从 \$\sqrt{2}\$ 谈起](#) 【获取码】SIGAI0620
- [27] [场景文本检测——CTPN 算法介绍](#) 【获取码】SIGAI0622
- [28] [卷积神经网络的压缩和加速](#) 【获取码】SIGAI0625
- [29] [k 近邻算法](#) 【获取码】SIGAI0627
- [30] [自然场景文本检测识别技术综述](#) 【获取码】SIGAI0627
- [31] [理解计算：从 \$\sqrt{2}\$ 到 AlphaGo ——第 2 季 神经计算的历史背景](#) 【获取码】SIGAI0704
- [32] [机器学习算法地图](#) 【获取码】SIGAI0706
- [33] [反向传播算法推导-全连接神经网络](#) 【获取码】SIGAI0709
- [34] [生成式对抗网络模型综述](#) 【获取码】SIGAI0709.
- [35] [怎样成为一名优秀的算法工程师](#) 【获取码】SIGAI0711.
- [36] [理解计算：从根号 2 到 AlphaGo——第三季 神经网络的数学模型](#) 【获取码】SIGAI0716
- [37] [【技术短文】人脸检测算法之 S3FD](#) 【获取码】SIGAI0716
- [38] [基于深度负相关学习的人群计数方法](#) 【获取码】SIGAI0718
- [39] [流形学习概述](#) 【获取码】SIGAI0723
- [40] [关于感受野的总结](#) 【获取码】SIGAI0723
- [41] [随机森林概述](#) 【获取码】SIGAI0725
- [42] [基于内容的图像检索技术综述——传统经典方法](#) 【获取码】SIGAI0727
- [43] [神经网络的激活函数总结](#) 【获取码】SIGAI0730
- [44] [机器学习和深度学习中值得弄清楚的一些问题](#) 【获取码】SIGAI0802

- [45] [基于深度神经网络的自动问答系统概述](#) 【获取码】SIGAI0803
- [46] [反向传播算法推导——卷积神经网络](#) 【获取码】SIGAI0806
- [47] [机器学习与深度学习核心知识点总结 写在校园招聘即将开始时](#) 【获取 码】SIGAI0808
- [48] [理解 Spatial Transformer Networks](#) 【获取码】SIGAI0810
- [49] [AI 时代大点兵-国内外知名 AI 公司 2018 年最新盘点](#) 【获取码】SIGAI0813
- [50] [理解计算：从 \$\sqrt{2}\$ 到 AlphaGo ——第 2 季 神经计算的历史背景](#) 【获取码】SIGAI0815
- [51] [基于内容的图像检索技术综述--CNN 方法](#) 【获取码】SIGAI0817
- [52] [文本表示简介](#) 【获取码】SIGAI0820
- [53] [机器学习中的最优化算法总结](#) 【获取码】SIGAI0822
- [54] [【AI 就业面面观】如何选择适合自己的舞台？](#) 【获取码】SIGAI0823
- [55] [浓缩就是精华-SIGAI 机器学习蓝宝石](#) 【获取码】SIGAI0824
- [56] [DenseNet 详解](#) 【获取码】SIGAI0827
- [57] [AI 时代大点兵国内外知名 AI 公司 2018 年最新盘点【完整版】](#) 【获取码】SIGAI0829
- [58] [理解 Adaboost 算法](#) 【获取码】SIGAI0831
- [59] [深入浅出聚类算法](#) 【获取码】SIGAI0903
- [60] [机器学习发展历史回顾](#) 【获取码】SIGAI0905
- [61] [网络表征学习综述](#) 【获取码】SIGAI0907
- [62] [视觉多目标跟踪算法综述\(上\)](#) 【获取码】SIGAI0910
- [63] [计算机视觉技术 self-attention 最新进展](#) 【获取码】SIGAI0912
- [64] [理解 Logistic 回归](#) 【获取码】SIGAI0914
- [65] [机器学习中的目标函数总结](#) 【获取码】SIGAI0917
- [66] [人脸识别中的活体检测算法综述](#) 【获取码】SIGAI0919
- [67] [机器学习与深度学习常见面试题（上）](#) 【获取码】SIGAI0921
- [68] [浅谈动作识别 TSN, TRN, ECO](#) 【获取码】SIGAI0924
- [69] [OCR 技术简介](#) 【获取码】SIGAI0926

- [70] [轻量化神经网络综述](#) 【获取码】SIGAI1001
- [71] [行人重识别 PCB-RPP, SGGNN](#) 【获取码】SIGAI1001
- [72] [人工智能非技术从业者必知的十件事](#) 【获取码】SIGAI1008
- [73] [理解生成模型与判别模型](#) 【获取码】SIGAI1012

原创声明：本文为 SIGAI 原创文章，仅供个人学习使用，未经允许，不能用于商业目的