目标检测算法综述

王荣胜
河南理工大学计算机科学与技术学院

摘要：目标检测（*Object Detection*）是计算机视觉领域的基本任务之一，在交通、医疗、工业生产等领域应用广泛。近年来，随着深度学习的快速发展和目标检测算法精度的不断提升，基于检测的多目标跟踪算法已超越传统跟踪方法，成为当前多目标跟踪算法的主流。文中首先回顾了在两个方向上目标跟踪算法的发展过程，之后将算法在各项指标上进行对比分析，并依据两类目标检测算法的优缺点对未来做出总结与展望。

关键词：目标检测；深度学习；

一、引言

目标检测是很多计算机视觉任务的基础，不论我们需要实现图像与文字的交互还是需要识别精细类别，它都提供了可靠的信息。目前目标检测领域的深度学习方法主要分为两类：*two stage* 的目标检测算法；*one stage* 的目标检测算法。前者是先由算法生成一系列作为样本的候选框，再通过卷积神经网络进行样本分类；后者则不用产生候选框，直接将目标边框定位的问题转化为回归问题处理。正是由于两种方法的差异，在性能上也有不同，前者在检测准确率和定位精度上占优，后者在算法速度上占优。

本文对目标检测进行了整体回顾，第一部分从 *RCNN*[1] 开始介绍基于候选区域的目标检测，包括 *Fast R-CNN*[3]、*Faster R-CNN*[4] 和 *R-FCN*[8] 等；第二部分则重点讨论了包括 *YOLO*[18]、*SSD*[21]和 *PPYOLO*[25] 等在内的单次检测器，它们都是目前最为优秀的方法。

二、基于候选区域的目标检测算法

在早期深度学习技术发展进程中，主要都是围绕分类问题展开研究，这是因为神经网络特有的结构输出将概率统计和分类问题结合，提供一种直观易行的思路。国内外研究人员虽然也在致力于将其他如目标检测领域和深度学习结合，但都没有取得成效，这种情况直到 R-*CNN* 算法出现才得以解决。

2.1 *R-CNN*

2014 年加州大学伯克利分校的 *Ross B. Girshick* 提出 *R-CNN* 算法，其在效果上超越同期的 *Yann Lecun* 提出的端到端方法 *OverFeat* 算法，其算法结构也成为后续 *two stage* 的经典结构。*R-CNN* 算法利用选择性搜索（*Selective Search*）算法评测相邻图像子块的特征相似度，通过对合并后的相似图像区域打分，选择出感兴趣区域的候选框作为样本输入到卷积神经网络结构内部，由网络学习候选框和标定框组成的正负样本特征，形成对应的特征向量，再由支持向量机设计分类器对特征向量分类，最后对候选框以及标定框完成边框回归操作达到目标检测的定位目的。虽然 *R-CNN* 算法相较于传统目标检测算法取得了 50%的性能提升，但其也有缺陷存在：训练网络的正负样本候选区域由传统算法生成，使得算法速度受到限制；卷积神经网络需要分别对每一个生成的候选区域进行一次特征提取，实际存在大量的重复运算，制约了算法性能。

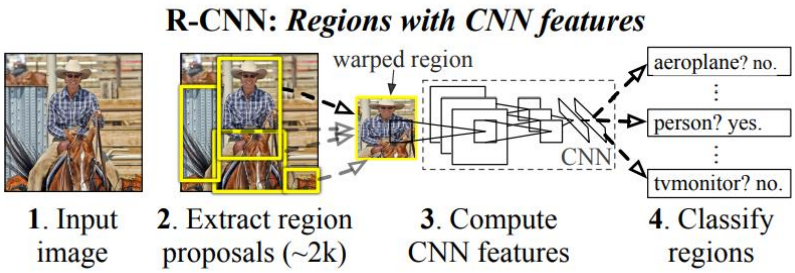

图 2.1 *R-CNN*

2.2 *SPP-Net*

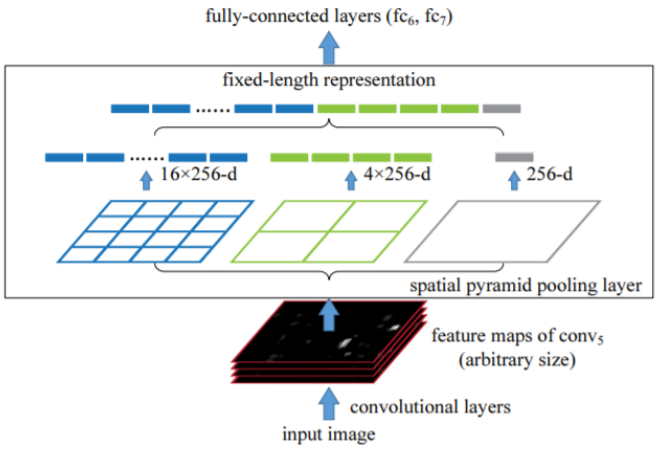

图 2.2 *spatial pyramid pooling layer*

针对卷积神经网络重复运算问题，2015 年微软研究院的何恺明等提出一种 *SPP-Net*[2]算法，通过在卷积层和全连接层之间加入空间金字塔池化结构（*Spatial Pyramid Pooling*）代替 *R-CNN* 算法在输入卷积神经网络前对各个候选区域进行剪裁、缩放操作使其图像子块尺寸一致的做法。利用空间金字塔池化结构有效避免了 *R-CNN* 算法对图像区域剪裁、缩放操作导致的图像物体剪裁不全以及形状扭曲等问题，更重要的是解决了卷积神经网络对图像重复特征提取的问题，大大提高了产生候选框的速度，且节省了计算成本。但是和 *R-CNN* 算法一样训练数据的图像尺寸大小不一致，导致候选框的 *ROI* 感受野大，不能利用 *BP* 高效更新权重。

2.3 *Fast R-CNN*

针对 *SPP-Net* 算法的问题，2015 年微软研究院的 *Ross B. Girshick* 又提出一种改进的 *Fast R-CNN* 算法，借鉴 *SPP-Net* 算法结构，设计一种 *ROI pooling* 的池化层结构，有效解决 *R-CNN* 算法必须将图像区域剪裁、缩放到相同尺寸大小的操作。提出多任务损失函数思想，将分类损失和边框回归损失结合统一训练学习，并输出对应分类和边框坐标，不再需要额外的硬盘空间来存储中间层的特征，梯度能够通过 *RoI Pooling* 层直接传播。但是其仍然没有摆脱选择性搜索算法生成正负样本候选框的问题。

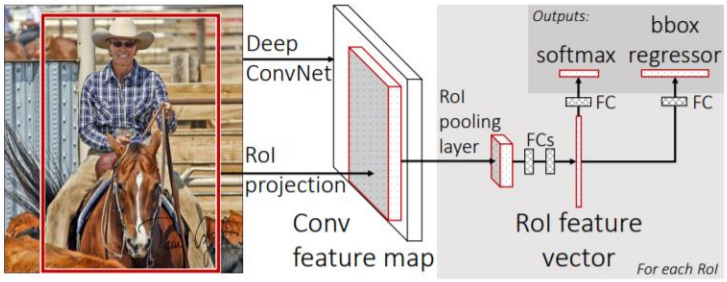

图 2.3 ***Fast R-CNN***

2.4 *Faster R-CNN*

为了解决 *Fast R-CNN* 算法缺陷，使得算法实现 *two stage* 的全网络结构，2015 年微软研究院的任少庆、何恺明以及 *Ross B Girshick* 等人又提出了 *Faster R-CNN* 算法。设计辅助生成样本的 *RPN*（*Region Proposal Networks*）网络，将算法结构分为两个部分，先由 *RPN* 网络判断候选框是否为目标，再经分类定位的多任务损失判断目标类型，整个网络流程都能共享卷积神经网络提取的的特征信息，节约计算成本，且解决 *Fast R-CNN* 算法生成正负样本候选框速度慢的问题，同时避免候选框提取过多导致算法准确率下降。但是由于 *RPN* 网络可在固定尺寸的卷积特征图中生成多尺寸的候选框，导致出现可变目标尺寸和固定感受野不一致的现象。

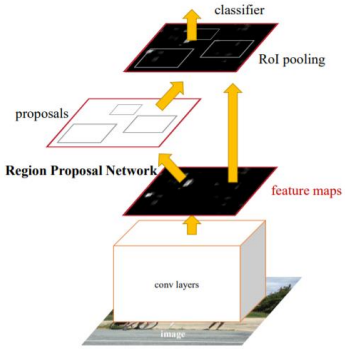

图 2.4 ***Faster R-CNN***

2.5 *MR-CNN*

2015 年巴黎科技大学提出 *MR-CNN*[5]算法，结合样本区域本身的特征，利用样本区域周围采样的特征和图像分割的特征来提高识别率，可将检测问题分解为分类和定位问题。

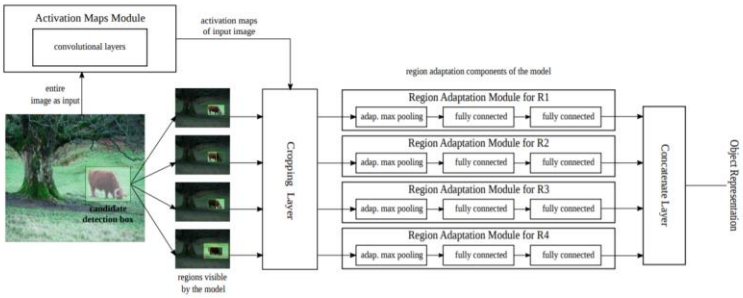

图 2.5 ***MR-CNN***

分类问题由 *Multi-Region CNN Model* 和 *Semantic Segmentation-Aware CNN Model* 组成。前者的候选框由 *Selective Search* 得到，对于每一个样本区域，取 10 个区域分别提取特征后拼接，这样可以强制网络捕捉物体的不同方面，同时可以增强网络对于定位不准确的敏感性，其中 *adaptive max pooling* 即 *ROI max pooling*；后者使用 *FCN* 进行目标分割，将最后一层的 *feature map* 和前者产生的 *feature map* 拼接，作为最后的 *feature map*。

为了精确定位，采用三种样本边框修正方法，分别为 *Bbox regression*、*Iterative localization* 以及 *Bounding box voting*。*Bbox regression*：在 *Multi-Region CNN Model* 中整幅图经过网路的最后一层卷积层后，接一个 *Bbox regression layer*，与 *RPN* 不同，此处的 *regression layer* 是两层 *FC* 以及一层 *prediction layer*，为了防止 *Selective Search* 得到的框过于贴近物体而导致无法很好的框定物体，将候选框扩大为原来的 1.3 倍再做。*Iterative localization*：初始的框是 *Selective Search* 得到的框，然后用已有的分类模型对框做出估值，低于给定阈值的框被筛掉，剩下的框用 *Bbox regression* 的方法调整大小，并迭代筛选。*Bounding box voting*：首先对经过 *Iterative localization* 处理后的框应用 *NMS*, $IOU=0.3$，得到检测结果，然后对于每一个框，用每一个和其同一类的而且 $IOU>0.5$ 的框加权坐标，得到最后的目标样本框。

2.6 *HyperNet*

2016 年清华大学提出 *HyperNet*[6]算法，其利用网络多个层级提取的特征，且从较前层获取的精细特征可以减少对于小物体检测的缺陷。将提取到的不同层级 *feature map* 通过最大池化降维或逆卷积扩增操作使得所有 *feature map* 尺寸一致，并利用 *LRN* 正则化堆叠，形成 *Hyper Feature maps*，其具有多层次抽象、合适分辨率以及计算时效性的优点。接着通过 *region proposal generation module* 结构进行预测和定位，仅保留置信度最高的 *N* 个样本框进行判断。

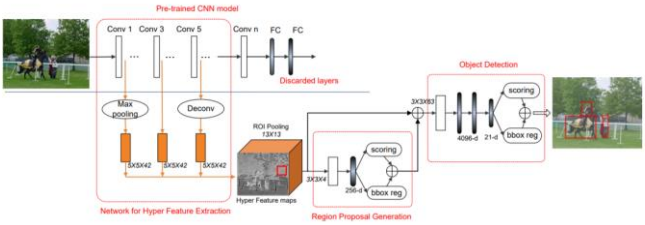

图 2.6 *HyperNet*

2.7 *CRAFT*

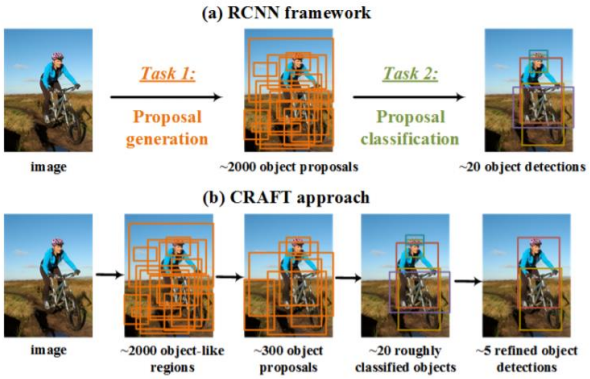

图 2.7 *CRAFT*

R-CNN 系列算法的第一阶段是生成目标 *proposals*，第二阶段是对目标 *proposals* 进行分类，2016 年中科院自动化所提出的 *CRAFT*[7]算法分别对 *Faster R-CNN* 中的这两个阶段进行了一定的改进。对于生成目标 *proposals* 阶段，在 *RPN* 的后面加了一个二值的 *Fast R-CNN* 分类器来对 *RPN* 生成的 *proposals* 进行进一步的筛选，留下一些高质量的 *proposals*；对于第二阶段的目标 *proposals* 分类，在原来的分类器后又级联了 *N* 个类别（不包含背景类）的二值分类器以进行更精细的目标检测。

2.8 *R-FCN*

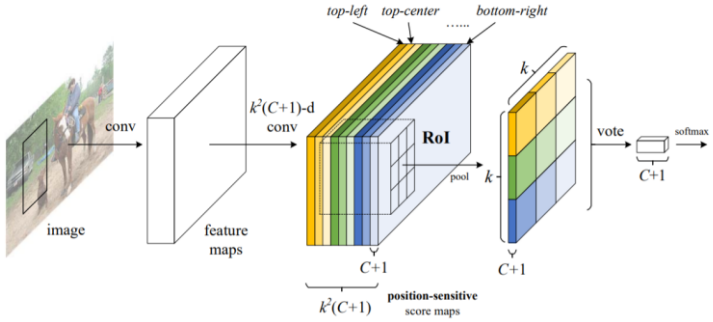

图 2.8 *R-FCN*

随着全卷积网络的出现，2016 年微软研究院的 *Jifeng Dai* 等提出 *R-FCN* 算法。相较于 *Faster R-CNN* 算法只能计算 *ROI pooling* 层之前的卷积网络特征参数，*R-FCN* 算法提出一种位置敏感分布的卷积网络代替 *ROI pooling* 层之后的全连接网络，解决了 *Faster R-CNN* 由于 *ROI Pooling* 层后面的结构需要对每一个样本区域跑一次而耗时比较大的问题，使得特征共享在整个网络内得以实现，解决物体分类要求有平移不变性和物体检测要求有平移变化的矛盾，但是没有考虑到 *region proposal* 的全局信息和语义信息。

2.9 *MS-CNN*

针对 *Faster R-CNN* 算法的遗留问题，2016 年加州大学圣地亚哥分校的 *Z Cai* 提出了 *MS-CNN*[9]算法，通过利用 *Faster R-CNN* 算法结构的多个不同层级输出的特征结果来检测目标，将不同层级的检测器互补形成多尺度的强检测器，应用浅层特征检测小尺寸目标，应用深层特征检测大尺寸目标。并且利用去卷积层代替图像上采样来增加图像分辨率，减少内存占用，提高运行速度。

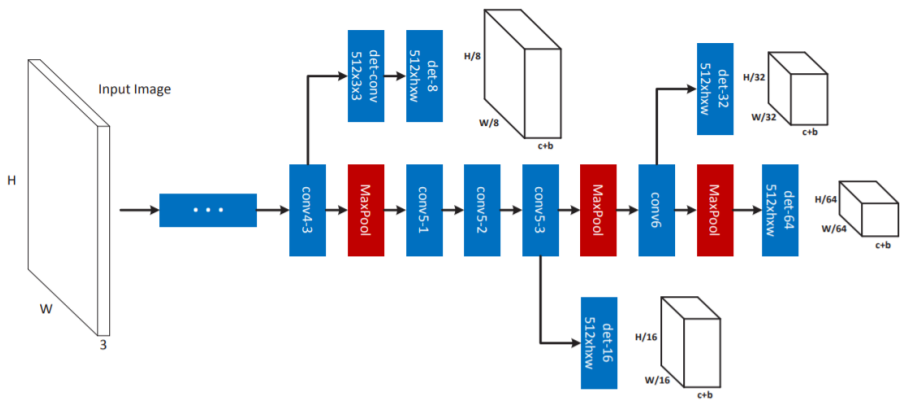

图 2.9 *MS-CNN* 特征网络结构

2.10 *PVANet*

针对的就是算法的运算速度提升问题，2016 年底 *Intel* 图像技术团队提出了一个轻量级的网络，取得了 *state-of-the-art* 的效果。*PVANet*[10]主要分为特征抽取网络和检测网络，基于多层少通道的基本原则，在网络浅层采用 *C.ReLU* 结构，在网络深层采用 *Inception* 模块，其中前者是将 *K*×*K* 卷积结构表示 *1*×*1* - *K*×*K* - *1*×*1* 的卷积层的堆叠，后者设计原则是由于为了检测图像中的大目标，需要足够大的感受野，可通过堆叠 *3*×*3* 的卷积核实现，但是为了捕获小目标，则需要小一点的感受野，可通过 *1*×*1* 的卷积核实现，且可以避免大卷积核造成的参数冗余问题。

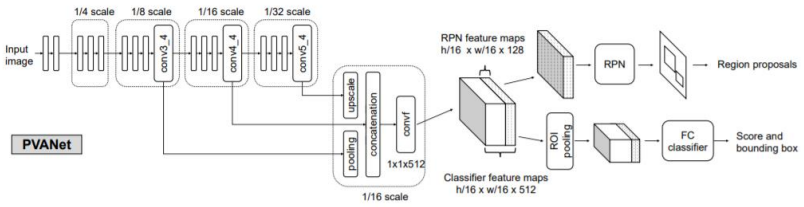

图 2.10 ***PVANet***

PVANet 应用多尺度特征级联最大化目标检测任务的多尺度性质，权重衰减策略采用一定迭代次数内 *loss* 不再下降，则将学习速率降低常数倍的方式，通过 *batch normalization* 和 *residual* 连接实现高效的训练。

2.11 *FPN*

2017 年 *Facebook* 的 *Tsung-Yi Lin* 等提出了 *FPN*[11]算法，利用不同层的特征图进行不同尺寸的目标预测。原来多数的目标检测算法都是只采用深层特征做预测，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。另外虽然也有些算法采用多尺度特征融合的方式，但是一般是采用融合后的特征做预测，而 *FPN* 算法不一样的地方在于预测是在不同特征层独立进行的，利用深层特征通过上采样和低层特征做融合。

FPN 算法主网络是 *ResNet*，结构主要是一个自底向上的线路横向连接一个自顶向下的线路。自底向上其实就是网络的前向过程，在前向过程中，*feature map* 的大小在经过某些层后会改变，而在经过其他一些层的时候不会改变，*FPN* 算法将不改变 *feature map* 大小的层归为一个 *stage*，因此每次抽取的特征都是每个 *stage* 的最后一个层输出，这样就能构成特征金字塔。自顶向下的过程采用上采样进行，而横向连接则是将上采样的结果和自底向上生成的相同大小的 *feature map* 并一一对应进行融合，在融合之后还会再采用 *3*×*3* 的卷积核对每个融合结果进行卷积，目的是消除上采样的混叠效应。

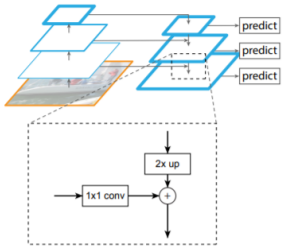

图 2.11 ***FPN***

2.12 *Mask R-CNN*

为了解决 *R-CNN* 算法为代表的 *two stage* 的方法问题，2017 年 *Facebook* 的何恺明等提出了 *Mask R-CNN*[12]算法，取得了很好的识别效果。*Mask R-CNN* 算法将 *ROI_Pooling* 层替换成了 *ROI_Align*，并且在边框识别的基础上添加分支 *FCN* 层（*mask* 层），用于语义 *Mask* 识别，通过 *RPN* 网络生成目标候选框，再对每个目标候选框分类判断和边框回归，同时利用全卷积网络对每个目标候选框预测分割掩膜。加入的掩膜预测结构解决了特征图像和原始图像上的 *ROI* 不对准问题，避免对 *ROI* 边界做任何量化，而用双线性插值到对准特征，再用池化操作融合。掩膜编码了输入图像的空间布局，用全卷积网络预测每个目标候选框的掩膜能完整的保留空间结构信息，实现目标像素级分割定位。

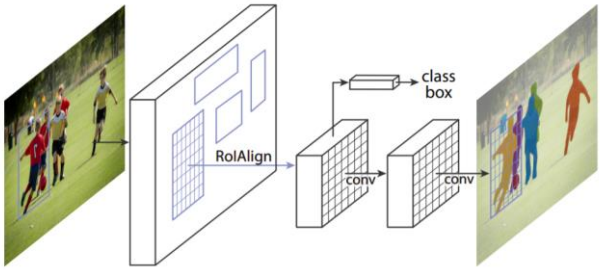

图 2.12 *Mask R-CNN*

2.13 *A-Fast-RCNN*

A-Fast-RCNN[13]算法是 2017 年卡内基梅隆大学提出的，其将对抗学习引入到目标检测问题中，通过对抗网络生成一下遮挡和变形的训练样本来训练检测网络，从而使得网络能够对遮挡和变形问题更加的鲁棒。使用对抗网络生成有遮挡和有形变的两种特征，两种网络分别为 *ASDN* 和 *ASTN*。

ASDN 利用 *Fast R-CNN* 中 *ROI* 池化层之后的每个目标 *proposal* 卷积特征作为对抗网络的输入，给定一个目标的特征，*ASDN* 尝试生成特征某些部分被 *dropout* 的掩码，导致检测器无法识别该物体。在前向传播过程中，首先使用 *ASDN* 在 *ROI* 池化层之后生成特征掩码，然后使用重要性采样法生成二值掩码，使用该掩码将特征对应部位值清零，修改后的特征继续前向传播计算损失，这个过程生成了困难的特征，用于训练检测器。

ASTN 主要关注特征旋转，定位网络包含三层全连接层，前两层是 *ImageNet* 预训练的 *FC6* 和 *FC7*，训练过程与 *ASDN* 类似，*ASTN* 对特征进行形变，将特征图划分为 4 个 *block*，每个 *block* 估计四个方向的旋转，增加了任务的复杂度。两种对抗网络可以相结合，使得检测器更鲁棒，*ROI* 池化层提取的特征首先传入 *ASDN* 丢弃一些激活，之后使用 *ASTN* 对特征进行形变。

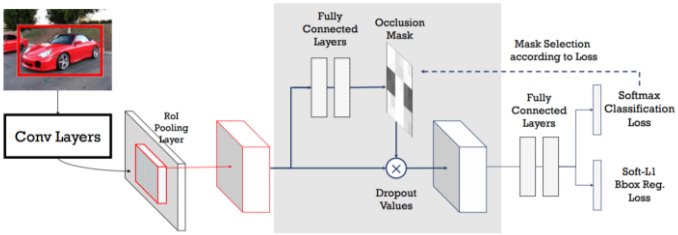

图 2.13 *A-Fast-RCNN*

2.14 *CoupleNet*

针对 *R-FCN* 算法没有考虑到 *region proposal* 的全局信息和语义信息的问题，2017 年中科院自动化所提出 *CoupleNet*[14]算法，其在原来 *R-FCN* 的基础上引入了 *proposal* 的全局和语义信息，通过结合局部、全局以及语义的信息，提高了检测的精度。

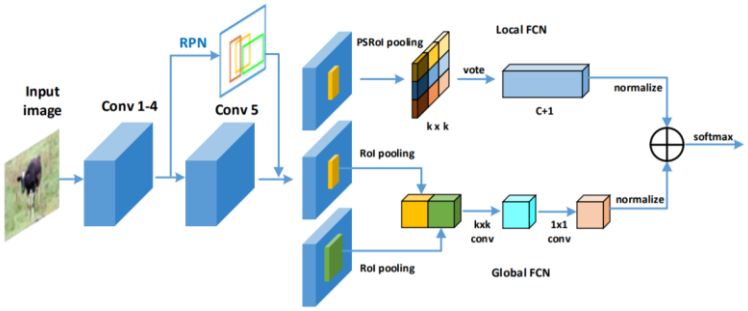

图 2.14 *CoupleNet*

CoupleNet 结构利用三支并行网络实现检测，上面的支路网络使用原本的 *R-FCN* 结构的位置敏感分布图提取目标的局部信息；中间的支路网络用于提取目标的全局信息，对于一个 *region proposal*，依次通过 *K*×*K* 的 *ROI Pooling*，*K*×*K* 的 *conv* 以及 *1*×*1* 的 *conv*；下面的支路网络用于提取目标的语义信息，对于一个 *region proposal*，首先选择以这个 proposal 为中心，面积是原来 2 倍的 *proposal*，同样依次通过 *K*×*K* 的 *ROI Pooling*，*K*×*K* 的 *conv* 以及 *1*×*1* 的 *conv*。最后先各自通过 *1*×*1* 的 *conv* 调整激活值的尺寸，然后把 *Local FCN* 和 *Global FCN* 结果对应位置元素相加，再通过一个 *softmax* 实现分类。

2.15 *MegDet*

基于 *CNN* 的物体检测研究一直在不断进步，从 *R-CNN* 到 *Fast/Faster R-CNN*，再 *Mask R-CNN*，主要的改进点都在于新的网络架构、新的范式、或者新的损失函数设计，然而 *mini-batch* 大小，这个训练中的关键因素并没有得到完善的研究。由于输入图片尺寸的增长，图像检测所需显存量也会同比例增长，这也使得已有的深度学习框架无法训练大 *mini-batch* 的图像检测模型，而小 *mini-batch* 的物体检测算法又常常会引入不稳定的梯度、*BN* 层统计不准确、正负样本比例失调以及超长训练时间的问题。因此，2017 年 12 月 *Face++*提出一种大 *mini-batch* 的目标检测算法 *MegDet*[15]。

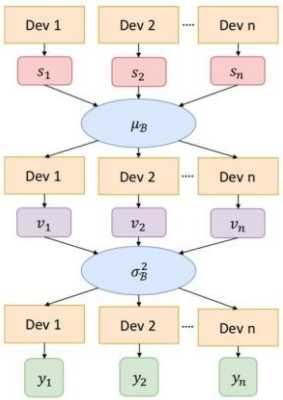

图 2.15 多 *GPU* 的 *Batch Normalization*

MegDet 算法可以使用远大于以往的 *mini-batch* 大小训练网络（比如从 16 增大到 256），这样同时也可以高效地利用多块 *GPU* 联合训练（在论文的实验中最多使用了 128 块 *GPU*），大大缩短训练时间。同时解决了 BN 统计不准确的问题，也提出了一种学习率选择策略以及跨 *GPU* 的 *Batch Normalization* 方法，两者共同使用就得以大幅度减少大 *mini-batch* 物体检测器的训练时间（比如从 33 小时减少到仅仅 4 个小时），同时还可以达到更高的准确率。

2.16 *Light-Head R-CNN*

2017 年 12 月 *Face++*提出了一种为了使 *two stage* 的检测算法 *Light-Head R-CNN*[16]，主要探讨了 *R-CNN* 如何在物体检测中平衡精确度和速度。*Light-Head R-CNN* 提出了一种更好的 *two-stage detector* 设计结构，使用一个大内核可分卷积和少量通道生成稀疏的特征图。该设计的计算量使随后的 *ROI* 子网络计算量大幅降低，检测系统所需内存减少。将一个廉价的全连接层附加到池化层上，充分利用分类和回归的特征表示。因其轻量级头部结构，该检测器能够实现速度和准确率之间的最优权衡，不管使用的是大主干网络还是小主干网络。

基于 *ResNet101* 网络达到了新的 *state-of-the-art* 的结果 40.6，超过了 *Mask R-CNN* 和 *RetinaNet*。同时如果是用一个更小的网络，比如类似 *Xception* 的小模型，达到了 100+*FPS*，30.7*mmap*，效率上超过了 *SSD* 和 *YOLO*。

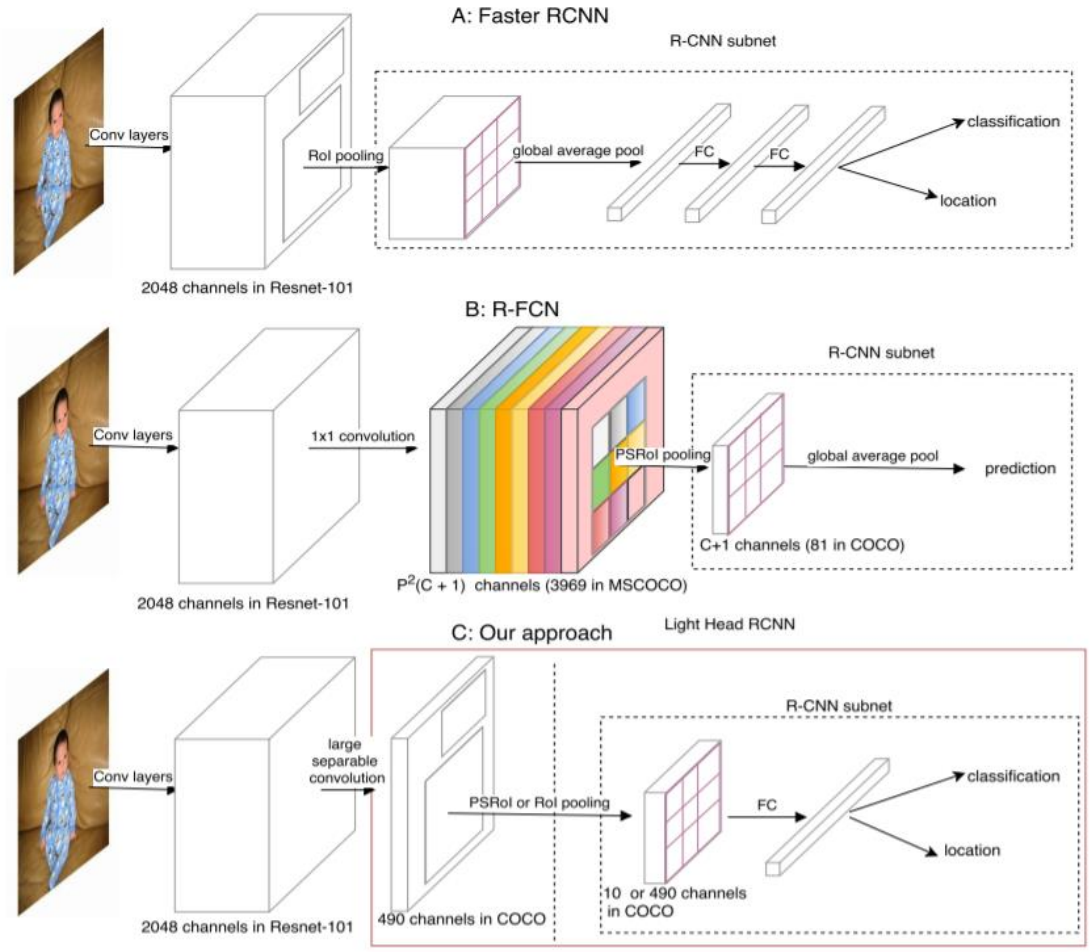

图 2.16 *Light-Head R-CNN*

三、单次检测器

以 *R-CNN* 算法为代表的 *two stage* 的方法由于 *RPN* 结构的存在，虽然检测精度越来越高，但是其速度却遇到瓶颈，比较难于满足部分场景实时性的需求。因此出现一种基于回归方法的 *one stage* 的目标检测算法，不同于 *two stage* 的方法的分步训练共享检测结果，*one stage* 的方法能实现完整单次训练共享特征，且在保证一定准确率的前提下，速度得到极大提升。

3.1 *OverFeat*

2013 年 *Yann Lecun* 在纽约大学的团队提出了著名的 *OverFeat*[17]算法，其利用滑动窗口和规则块生成候选框，再利用多尺度滑动窗口增加检测结果，解决图像目标形状复杂、尺寸不一问题，最后利用卷积神经网络和回归模型分类、定位目标。该算法首次将分类、定位以及检测三个计算机视觉任务放在一起解决，获得同年 *ILSVRC 2013* 任务 3（分类+定位）的冠军，但其很快就被同期的 *R-CNN* 算法取代。

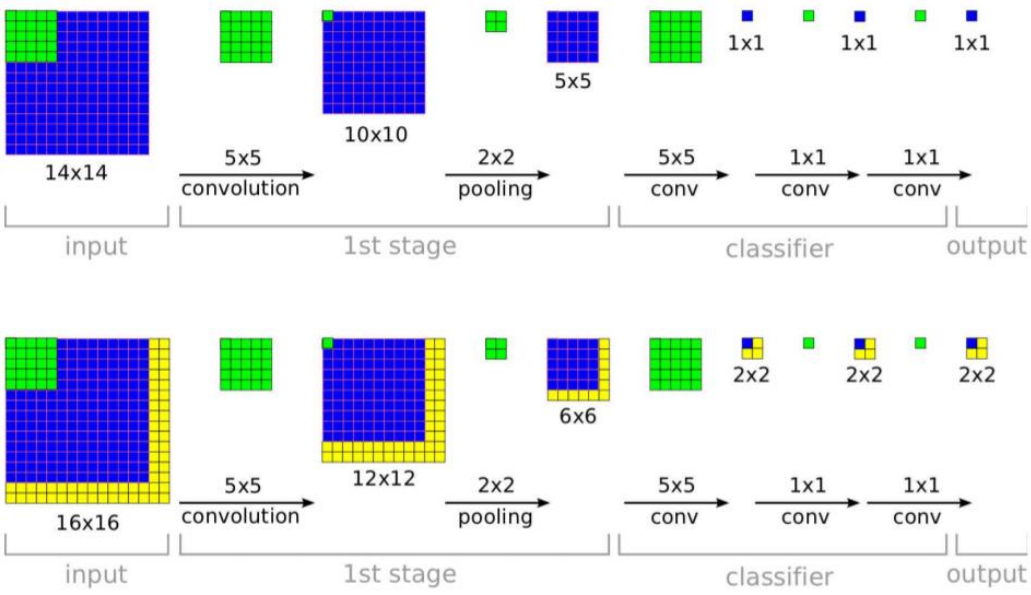

图 3.1 用于检测的高效卷积

3.2 *YOLO*

2015 年华盛顿大学的 *Joseph Redmon* 等提出的 *YOLO* 算法继承了 *OverFeat* 算法这种基于回归的 *one stage* 方法，速度能达到每秒 45 帧，由于其速度优势迅速成为端到端方法的领先者。*YOLO* 算法是基于图像的全局信息进行预测的，整体结构简单，通过将输入图像重整到 *448×448* 像素固定尺寸大小，并划分图像为 *7×7* 网格区域，通过卷积神经网络提取特征训练，直接预测每个网格内的边框坐标和每个类别置信度，训练时采用 *P-Relu* 激活函数。但是存在定位不准以及召回率不如基于区域提名方法的问题，且对距离很近的物体和很小的物体检测效果不好，泛化能力相对较弱。

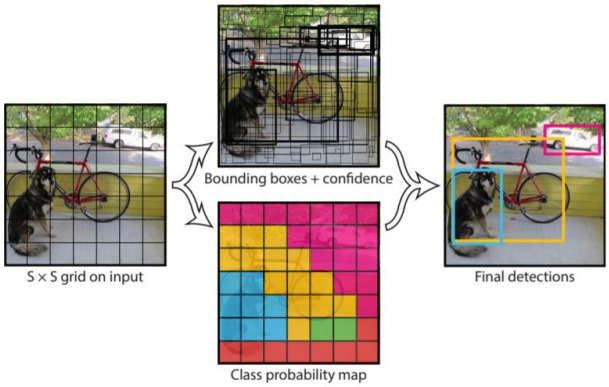

图 2 *YOLO*

3.3 *YOLOv2 & YOLO9000*

经过 *Joseph Redmon* 等的改进，*YOLOv2* 和 *YOLO9000*[19]算法在 2017 年 *CVPR* 上被提出，并获得最佳论文提名，重点解决召回率和定位精度方面的误差。采用 *Darknet-19* 作为特征提取网络，增加了批量归一化（*Batch Normalization*）的预处理，并使用 224×224 和 448×448 两阶段训练 *ImageNet* 预训练模型后 *fine-tuning*。相比于原来的 *YOLO* 是利用全连接层直接预测 *bounding box* 的坐标，*YOLOv2* 借鉴了 *Faster R-CNN* 的思想，引入 *anchor* 机制，利用 *K-Means* 聚类的方式在训练集中聚类计算出更好的 *anchor* 模板，在卷积层使用 *anchor boxes* 操作，增加候选框的预测，同时采用较强约束的定位方法，大大提高算法召回率。结合图像细粒度特征，将浅层特征与深层特征相连，有助于对小尺寸目标的检测。

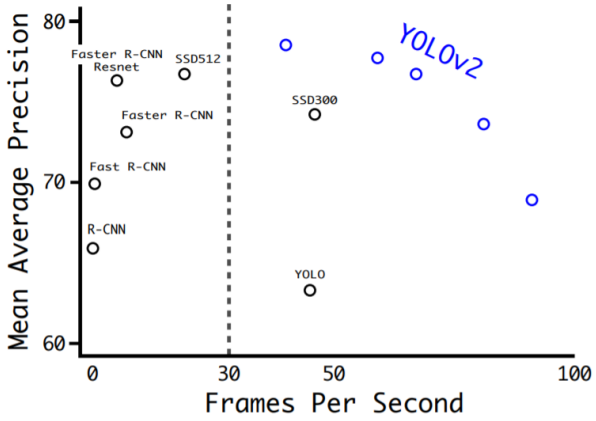

图 3.3 *YOLOv2* 在 *VOC2007* 上的速度和精度

3.4 *G-CNN*

由于巨大的 *proposal* 数量使得后续检测效率降低，2016 年马里兰大学的 *M Najibi* 等提出一种起始于网格迭代的 *G-CNN*[20]算法。通过初始化对图像划分回归后得到更加接近物体的候选框，再利用回归框作为原始窗口进行回归调整，解决了以往的基于区域提名方法通过海量潜在候选框直接进行目标搜索，抑制负样本的缺陷。

在训练阶段，首先在图像中获取叠加的多尺度的规则网格（实际网格相互叠加），然后通过 *ground truth* 与每一个网格的 *IOU* 进行每一个网格 *ground truth* 的分配，并完成训练过

程，使得网格在回归过程中渐渐接近 *ground truth*。在检测阶段，对于每一个样本框针对每一类获得置信分数，用最可能类别的回归器来更新样本框的位置。

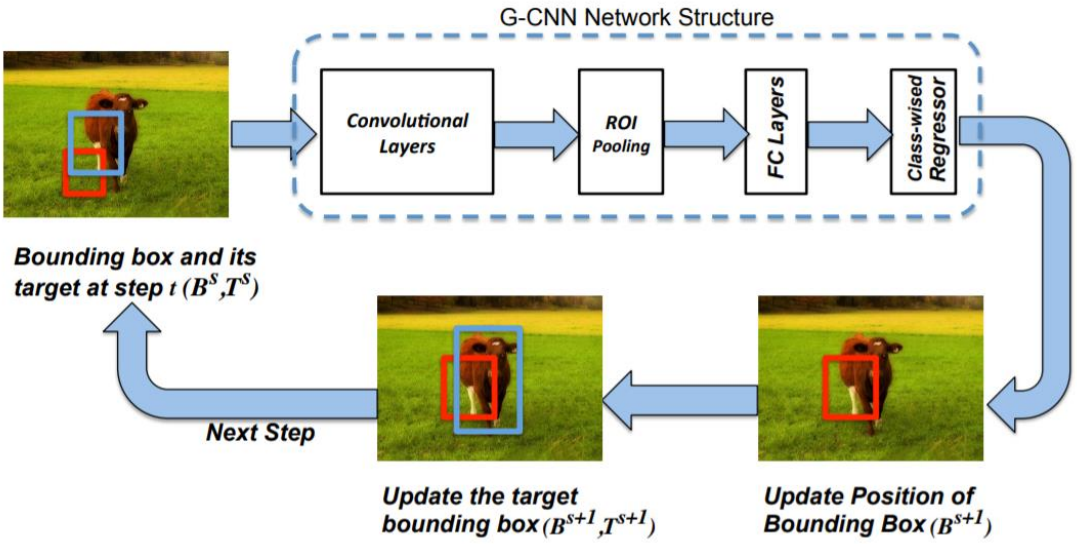

图 3.4 *G-CNN*

3.5 *SSD*

针对 *YOLO* 类算法的定位精度问题，2016 年 12 月北卡大学教堂山分校的 *Wei Liu* 等提出 *SSD* 算法，将 *YOLO* 的回归思想和 *Faster R-CNN* 的 *anchor box* 机制结合。通过在不同卷积层的特征图上预测物体区域，输出离散化的多尺度、多比例的 *default boxes* 坐标，同时利用小卷积核预测一系列候选框的边框坐标补偿和每个类别的置信度。在整幅图像上各个位置用多尺度区域的局部特征图边框回归，保持 *YOLO* 算法快速特性的同时，也保证了边框定位效果和 *Faster R-CNN* 类似。但因其利用多层次特征分类，导致其对于小目标检测困难，最后一个卷积层的感受野范围很大，使得小目标特征不明显。

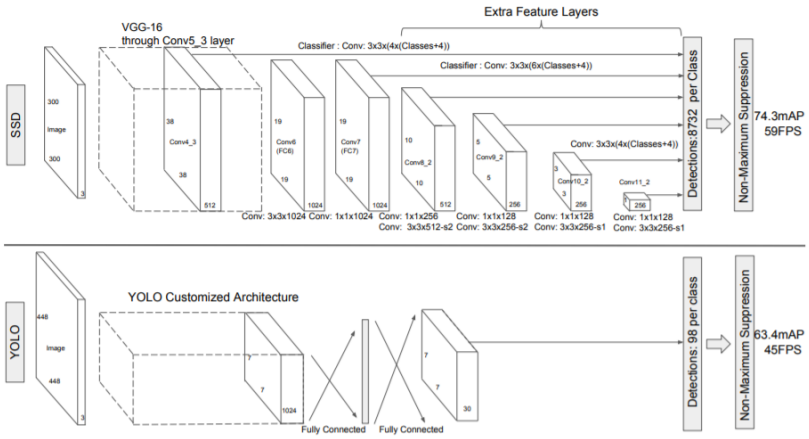

图 3.5 *SSD* 和 *YOLO* 网络结构对比

3.6 *R-SSD*

2017 年首尔大学提出了 *R-SSD*[21]算法，解决了 *SSD* 算法中不同层 *feature map* 都是独立

作为分类网络的输入，容易出现相同物体被不同大小的框同时检测出来的情况，还有对小尺寸物体的检测效果比较差的情况。*R-SSD* 算法一方面利用分类网络增加不同层之间的 *feature map* 联系，减少重复框的出现；另一方面增加 *feature pyramid* 中 *feature map* 的个数，使其可以检测更多的小尺寸物体。特征融合方式采用同时利用 *pooling* 和 *deconvolution* 进行特征融合，这种特征融合方式使得融合后每一层的 *feature map* 个数都相同，因此可以共用部分参数，具体来讲就是 *default boxes* 的参数共享。

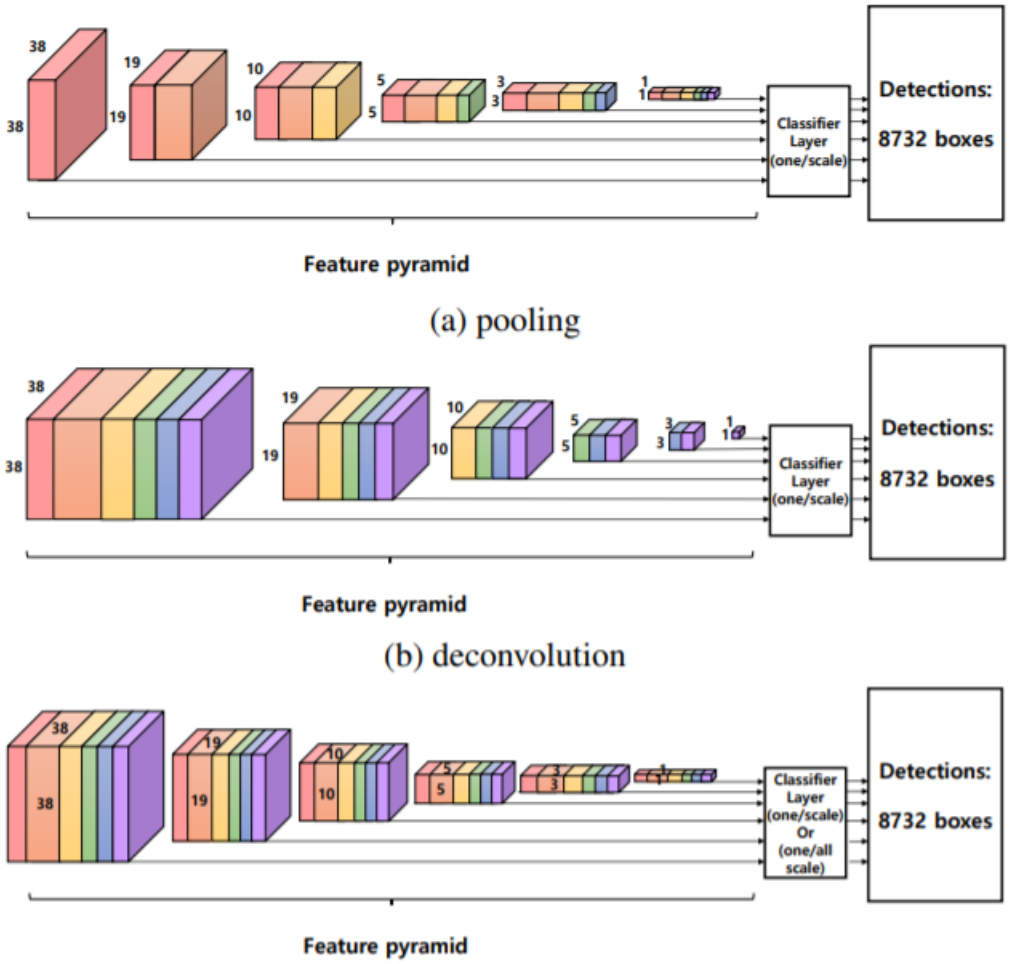

图 3.6 三种特征融合方式

3.7 *DSSD*

为了解决 *SSD* 算法检测小目标困难的问题，2017 年北卡大学教堂山分校的 *Cheng-Yang Fu* 等提出 *DSSD*[22]算法，将 *SSD* 算法基础网络从 *VGG-16* 更改为 *ResNet-101*，增强网络特征提取能力，其次参考 *FPN* 算法思路利用去卷积结构将图像深层特征从高维空间传递出来，与浅层信息融合，联系不同层级之间的图像语义关系，设计预测模块结构，通过不同层级特征之间融合特征输出预测物体类别信息。

DSSD 算法中有两个特殊的结构：*Prediction* 模块；*Deconvolution* 模块。前者利用提升每个子任务的表现来提高准确性，并且防止梯度直接流入 *ResNet* 主网络。后者则增加了三个 *Batch Normalization* 层和三个 *3×3* 卷积层，其中卷积层起到了缓冲的作用，防止梯度对主网络影响太剧烈，保证网络的稳定性。

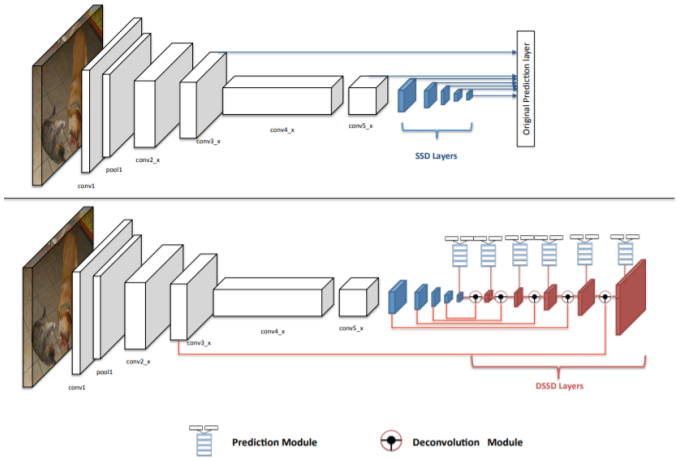

图 3.7 *SSD* 和 *DSSD* 网络结构对比

3.8 *DSOD*

2017 年复旦大学提出 *DSOD*[23]算法，其并不是在 *mAP* 上和其他检测算法做比较，看谁的算法更有效或者速度更快，而是从另一个角度切入说明 *fine-tune* 和直接训练检测模型的差异其实是可以减小的，也就是说训练一个检测模型可以不需要大量的数据和预训练好的模型。这是由于预训练模型的限制导致：迁移模型结构灵活性差，难以改变网络结构；分类任务预训练模型和检测任务训练会有学习偏差；虽然微调会减少不同目标类别分布的差异性，但深度图等特殊图像迁移效果差异较大。

SSD 算法是在六个尺度的特征图上进行检测，将这六个检测结果综合起来，*DSOD* 算法则则根据 *DenseNet* 的设计原理，将相邻的检测结果一半一半的结合起来。*DSOD* 算法是基于 *SSD* 算法基础上做的修改，采用的特征提取网络是 *DenseNet*。采用 *Dense Block* 结构，能避免梯度消失的情况。同时利用 *Dense Prediction* 结构，也能大大减少模型的参数量，特征包含更多信息。设计 *stem* 结构能减少输入图片信息的丢失，*stem* 结构由 *3×3* 卷积和 *2×2* 的 *max pool* 层组成，其还可以提高算法检测的 *mAP*。

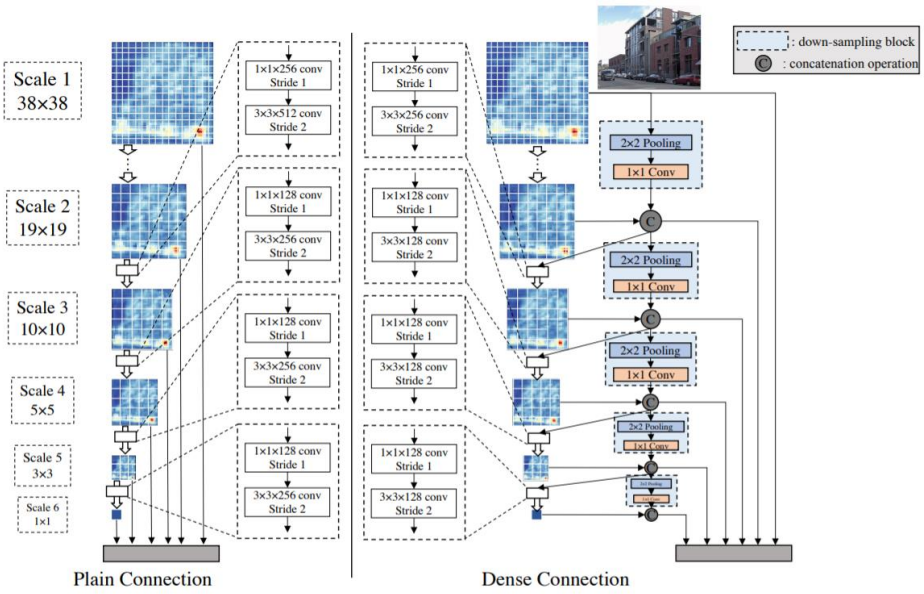

图 3.8 *DSOD* 预测层

3.9 *RON*

2017 年清华大学提出了 *RON*[24]算法，结合 *two stage* 名的方法和 *one stage* 方法的优势，更加关注多尺度对象定位和负空间样本挖掘问题。

- 多尺度对象定位——各种尺度的物体可能出现在图像的任何位置，因此应考虑成千上万个具有不同位置/尺度/方位的区域。多尺度表征将显著改善各种尺度的物体检测，但是这些方法总是在网络的一层检测到各种尺度的对象；
- 负空间挖掘——对象和非对象样本之间的比例严重不平衡。因此，对象检测器应该具有有效的负挖掘策略。

RON 算法通过设计方向连接结构，利用多尺度表征显著改善各种多尺度物体检测，同时为了减少对象搜索空间，在卷积特征图创建 *objectness prior* 引导目标对象搜索，训练时将检测器进行联合优化。并通过多任务损失函数联合优化了反向连接、*objectness prior* 和对象检测，因此可直接预测各种特征图所有位置的最终检测结果。

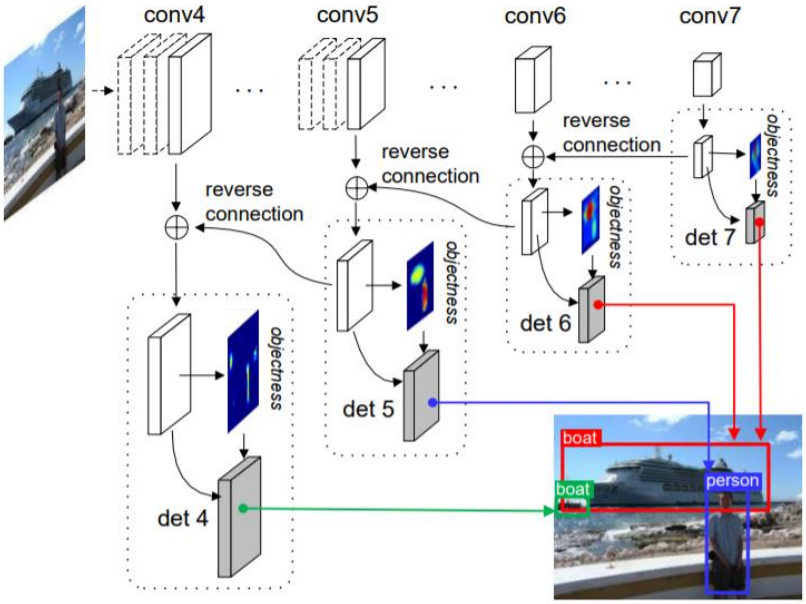

图 3.9 *RON*

3.10 *PPYOLO*

PP-YOLO 的命名规则很简单，因为作者团队是百度，而 *PaddlePaddle* 就是百度开源的深度学习框架，所以其全称是 *PaddlePaddle-YOLO*，简称即 *PP-YOLO*。

- *PP-YOLO* 的目的是实现一种可以在实际应用场景中直接应用的具有相对平衡的有效性和效率的目标检测器，而不是提出一种新颖的检测模型。
- *PP-YOLO*：一种基于 *YOLOv3* 的新型目标检测器。
- *PP-YOLO*：尝试结合各种几乎不增加模型参数和 *FLOPs* 数量的技巧，以实现在确保速度几乎不变的情况下尽可能提高检测器精度的目标。

PP-YOLO 则是在 *YOLOv3* 的基础上加了一些 *tricks*，相比于 *YOLOv4*，*PPYOLO* 没有在骨干网络和数据增强上做变换，拓展性还是较强的。

YOLO 系列的内容在工业界被广泛使用，截止至这篇论文完成之时，更新到 *YOLOv4*。

YOLOv4 提到了巨量的 *tricks* 分成 *bag of freebies* 和 *bag of specials* 分别用于训练阶段和测试阶段。

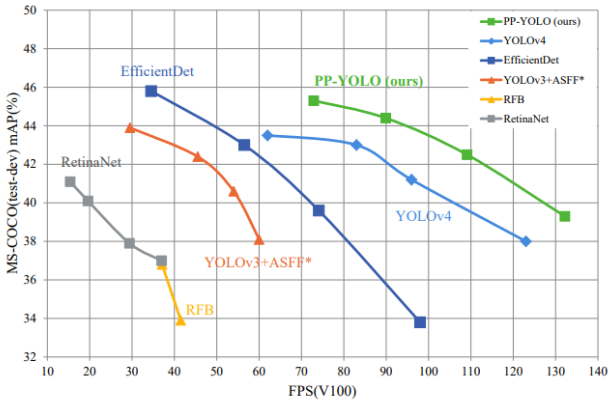

图 3.10.1 *PPYOLO* 性能指标与其它模型比较

PP-YOLO 是 *PaddleDetection* 优化和改进的 *YOLOv3* 的模型，其精度(*COCO* 数据集 *mAP*)和推理速度均优于 *YOLOv4* 模型，*PP-YOLO* 在 *COCO test-dev2017* 数据集上精度达到 *45.9%*，在单卡 *V100* 上 *FP32* 推理速度为 *72.9 FPS*, *V100* 上开启 *TensorRT* 下 *FP16* 推理速度为 *155.6 FPS*。

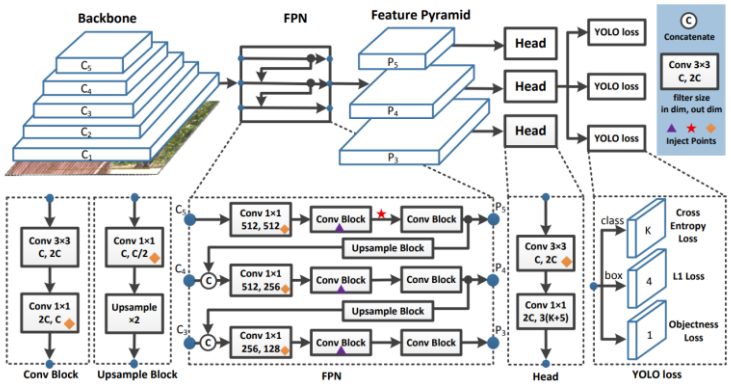

图 3.10.2 一个单阶段的 *Anchor based* 的检测模型通常是由一个骨架网络一个 *neck*（通常是 *FPN*），以及一个 *head*（用于分类+定位）组成。*PP-YOLO* 选择了 *ResNet50-vd-dcn* 作为骨架网络。

3.11 *YOLOv5*

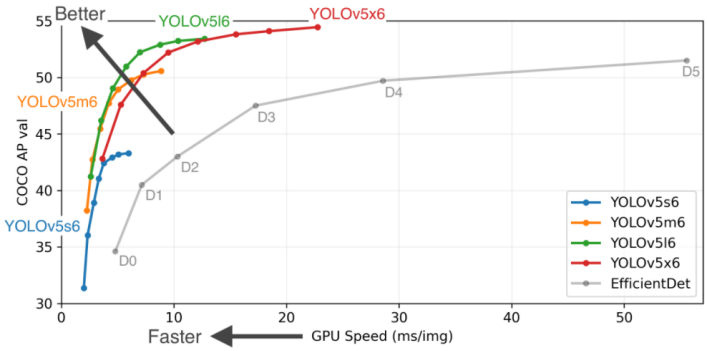

图 3.11 *YOLOv5* 四个网络性能对比

YOLO v5 和前 *YOLO* 系列相比，特点应该是：

（1）增加了正样本：方法是邻域的正样本 *anchor* 匹配策略；

（2）通过灵活的配置参数，可以得到不同复杂度的模型；

（3）通过一些内置的超参优化策略，提升整体性能；

（4）和 *yolov4* 一样，都用了 *mosaic* 增强，提升小物体检测性能。

四、总结与展望

随着深度学习技术在图像各领域的研究深入，出现越来越多的新理论、新方法。*two stage* 的方法和基于回归思想的 *one stage* 方法两者相互借鉴，不断融合，取得了很好的效果，也为我们展示了一些未来发展趋势：

- 参考上下文特征的多特征融合；
- 多尺度的对象定位；
- 结合循环神经网络（*RNN*）的图像语义分析。

算法	网络	测试图像尺寸	VOC 2007	VOC 2010	VOC 2012	ILSVRC 2013	MSCOCO 2015	速度
OverFeat						24.3%		
R-CNN	AlexNet		58.5%	53.7%	53.3%	31.4%		
R-CNN	VGG16		66.0%					
SPP-Net	ZF-Net		54.2%			31.84%		
Fast R-CNN	VGG16		70.0%	68.8%	68.4%		19.7%	
Faster R-CNN	VGG16		78.8%		75.9%		21.9%	
Faster R-CNN	ResNet101		85.6%		83.8%		37.4%	198ms
MR-CNN			73.9%					
YOLO			63.4%		57.9%			45fps
YOLO	VGG16		66.4%					21fps
YOLOv2		448×448	78.6%		73.4%		21.6%	40fps
SSD	VGG16	300×300	77.2%		75.8%		25.1%	46fps
SSD	VGG16	512×512	79.8%		78.5%		28.8%	19fps
SSD	ResNet101	300×300					28.0%	16fps
SSD	ResNet101	512×512					31.2%	8fps
DSSD	ResNet101	300×300					28.0%	8fps
DSSD	ResNet101	500×500					33.2%	6fps
CRAFT			75.7%		71.3%	48.5%		
OHEM			78.9%		76.3%		25.5%	
R-FCN	ResNet50		77.4%					0.09sec
R-FCN	ResNet101		79.5%					0.12sec
PVANet			84.9%		84.2%			46ms
Light-Head R-CNN	Xception	800×1200					31.5%	95fps
Light-Head R-CNN	Xception	700×1100					30.7%	102fps

图 4.1 经典目标检测算法在基准数据上的性能指标比对(最新的 *PPYOLO* 和 *YOLOv5* 未纳入比对)

两类算法都有其适用的范围，比如说实时快速动作捕捉，*one stage* 更胜一筹；复杂、多物体重叠，*two stage* 当仁不让。没有不好的算法，只有合适的使用场景。

参考文献

[1]*Ross Girshick，Rich feature hierarchies for accurate object detection and semantic segmentation.*

[2] *Kaiming He，Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.*

[3] *Ross Girshick，Fast R-CNN.*

[4] *Shaoqing Ren，Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.*

[5] *Spyros Gidaris，Object detection via a multi-region & semantic segmentation-aware CNN model.*

[6] *Tao Kong，HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection.*

[7] *Bin Yang，CRAFT Objects from Images.*

[8] *Jifeng Dai，R-FCN: Object Detection via Region-based Fully Convolutional Networks.*

[9] *Zhaowei Cai，A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection.*

[10] *Sanghoon Hong，PVANet: Lightweight Deep Neural Networks for Real-time Object Detection.*

[11] *Tsung-Yi Lin，Feature Pyramid Networks for Object Detection.*

[12] *Kaiming He，Mask R-CNN.*

[13] *Xiaolong Wang，A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection.*

[14] *Yousong Zhu，CoupleNet: Coupling Global Structure with Local Parts for Object Detection.*

[15] *Chao Peng ，MegDet: A Large Mini-Batch Object Detector.*

[16] *Zeming Li，Light-Head R-CNN: In Defense of Two-Stage Object Detector.*

[17] *Xiang Zhang ，OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks.*

[18] *Joseph Redmon ，YOLO9000: Better, Faster, Stronger.*

[19] *Mahyar Najibi ，G-CNN: an Iterative Grid Based Object Detector.*

[20] *Wei Liu ，SSD: Single Shot MultiBox Detector.*

[21] *Jisoo Jeong ，Enhancement of SSD by concatenating feature maps for object detection.*

[22] *Cheng-Yang Fu，DSSD : Deconvolutional Single Shot Detector.*

[23] *Zhiqiang Shen ，DSOD: Learning Deeply Supervised Object Detectors from Scratch.*

[24] *Tao Kong ，RON: Reverse Connection with Objectness Prior Networks for Object Detection.*

[25] *PP-YOLO: An Effective and Efficient Implementation of Object Detector.*