基于正负锚点框均衡及特征对齐的单阶段目标检测算法

唐乾坤[1,2)], 胡瑜[1,2)*]

[1)] (中国科学院计算技术研究所智能计算机研究中心 北京 100190)
[2)] (中国科学院大学 北京 100049)
(huyu@ict.ac.cn)

摘 要: 针对正负例锚点框不均衡将降低基于锚点框的单阶段目标检测算法的检测精度的问题, 提出一种包含锚点框提升模块和特征对齐模块来均衡正负例锚点框的算法. 首先在锚点框提升模块中预测各个锚点框为正例的可能性, 并粗略调整初始锚点框的位置和尺寸; 然后在特征对齐模块中为调整后的锚点框提取预测所需的对齐特征; 最后检测网络借助锚点框提升模块输出信息, 从调整后的锚点框中识别出简单负例锚点框, 并在训练阶段忽略其梯度. 将文中算法应用于以 VGG-16 和 ResNet-101 为特征提取网络的编解码架构中, 在目标检测数据集 MS COCO 和 PASCAL VOC 上进行实验, 结果表明, 该算法能够显著改善不均衡问题, 提高单阶段目标检测算法的检测精度(MS COCO 和 PASCAL VOC 上的精度分别为 42.8%和 82.7%), 并维持 28.6 帧/s 的实时运行速度.

关键词: 卷积神经网络; 单阶段目标检测; 锚点框正负例不均衡; 锚点框提升模块; 特征对齐模块
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2020.18175

PosNeg-Balanced Anchors with Aligned Features for Single-Shot Object Detection

Tang Qiankun[1,2)] and Hu Yu[1,2)*]

[1)] (*Research Center for Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)
[2)] (*University of Chinese Academy of Sciences, Beijing* 100049)

Abstract: We introduce a novel single-shot object detector to ease the imbalance of foreground-background class by suppressing the easy negatives while increasing the positives. To achieve this, we propose an anchor promotion module (APM) which predicts the probability of each anchor as positive and adjusts their initial locations and shapes to promote both the quality and quantity of positive anchors. In addition, we design an efficient feature alignment module to extract aligned features for fitting the promoted anchors with the help of both location and shape transformation information from the APM. The probabilities from APM are helpful for the detection classifier to identify the easy negatives and to ignore their gradients during training. We assemble the proposed modules to the backbone of VGG-16 and ResNet-101 network with an encoder-decoder architecture. Extensive experiments on MS COCO and PASCAL VOC well demonstrate our model performs competitively with alternative methods (42.8% mAP on MS COCO and 82.7% mAP on PASCAL VOC) and can run at 28.6 FPS.

Key words: convolution neural network; single-shot object detection; imbalance of foreground-background anchors; anchor promotion module; feature alignment module

收稿日期: 2019-12-10; 修回日期: 2020-03-31. **基金项目**: 国家重点研发计划科技创新 2030—"新一代人工智能"重大项目(2018AAA0102701); 空间智能控制技术实验室开放基金(HTKJ2019KL502003); 中国科学院计算技术研究所创新课题(20186090).
唐乾坤(1993—), 男, 博士研究生, 主要研究方向为计算机视觉、目标检测; **胡瑜**(1975—), 女, 博士, 研究员, 博士生导师, CCF 会员, 论文通讯作者, 主要研究方向为自主导航、自动驾驶感知与决策、深度学习算法加速.

深度学习的发展极大地促进了目标检测算法的进步. 基于深度学习的目标检测算法大致可分为两阶段目标检测算法和单阶段目标检测算法. 两阶段目标检测算法[1-6]首先生成稀疏的区域建议框, 然后使用一个子网络进一步调整区域建议框的位置并给出类别. 两阶段目标检测算法的检测精度通常较高, 但运行速度较慢(往往低于 10 帧/s). 单阶段目标检测算法直接根据预先划分的网格[7]或者设置的锚点框[8-9]预测出边界框, 因而运行速度快(往往高于 15 帧/s). 基于锚点框的单阶段检测算法, 其检测精度通常不及两阶段目标检测算法, 训练时的锚点框不均衡则是造成检测精度下降的重要原因.

本文首先分析和总结现有基于锚点框的单阶段目标检测算法在训练阶段存在的 3 种不均衡性及相关工作的优缺点; 为此提了一个锚点框提升模块(anchor promotion module, APM), 其调整锚点框以缓解数量和定位质量不均衡; 其输出锚点框是正例的概率信息, 以利于识别并抑制简单负例样本, 缓解分类难易不均衡; 其次提了一个特征对齐模块(feature alignment module, FAM), 为调整后的锚点框提取对齐的特征表达, 有利于提高检测网络的预测精度; 最后在公开的目标检测数据集(MS COCO 和 PASCAL VOC)上评估本文算法解决不均衡问题的有效性, 并与同类算法在检测精度和运行速度方面进行比较.

1 锚点框不均衡问题

锚点框是单阶段目标检测算法重要的组件之一. 锚点框能够减少检测算法搜索目标的范围, 使其快速地定位目标位置. 但是本文观察发现, 训练单阶段目标检测算法时的锚点框不均衡影响了最终的检测精度. 这种不均衡表现在 3 个方面.

(1) 数量不均衡, 即背景类的锚点框(负例样本)数量远远多于前景类的锚点框(正例样本)数量. 为了提高检测结果的召回率(average recall, AR), 基于锚点框的单阶段检测算法需要在用于预测的特征图的各个位置设置不同尺度和大小的锚点框. 由于一张图片往往只有几个物体存在, 背景占据了图片的主体部分, 因此该操作引入了大量的负例锚点框. 如图 1“初始”所示, 负例锚点框平均有 18 198.3 个, 而正例锚点框平均只有 28.4 个(具体实验设置见第 3.1 节和第 3.2 节).

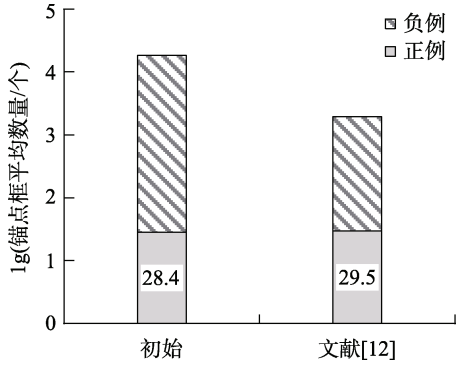

图 1 不同设置下正例与负例锚点框平均数量

为解决正负锚点框数量的不均衡, 两阶段目标检测算法[3-5]首先生成稀疏的区域建议框, 并从稀疏的建议框中启发式地采样一定比值(比如 1:3)的正负例样本进行训练, 启发式采样只能在一定程度上缓解正负例样本数量的不均衡. 该类算法仅考虑训练样本子集内部的数量均衡性, 并未考虑全部样本. 对于单阶段目标检测算法, 文献[10]使用一个先验模块预测某个位置有物体的可能性, 以减少后续阶段的搜索范围; 文献[11]设计了一个锚点框精调模块对预先设置的锚点框进行二元分类(包含物体与否), 并以固定的阈值减少易识别的样本数量, 训练检测网络时使用难负样本挖掘 1:3 正负样本; 文献[12]选择性地减少低层检测层的样本数量, 如图 1 所示. 这些工作仅使用锚点框包含物体的可能性(Objectness)减少样本数量, 由于 Objectness 并非完全准确, 不仅负例样本数量减少, 正例样本数量也会减少.

(2) 分类难易不均衡, 即样本被正确识别的难易程度不同. 由于图片中物体的大小和环境等因素的差异, 以及用于锚点框预测的特征表达能力的差异, 某些样本能被轻易识别(即易分类), 而有些样本会产生歧义类别(即难分类), 且易分类样本数量远远多于难分类样本. 对分类器而言, 易分类样本提供的有用信息较少, 而且数量过多的易分类样本的梯度会误导分类器的训练.

文献[13]提出了 Focal loss 算法, 使用一个调节因子以衰减已被很好分类样本的损失权重, 使分类器更多地关注难样本. 文献[14]从样本的梯度范数视角提出根据梯度范数在一定范围内的样本数量决定损失权重, 以减少易分类样本和异常样本的损失权重. 但上述文献并没有明确的标准定义难易样本. 尽管已被很好分类样本的损失权重被降低, 但是其梯度依然存在, 仍会影响分类器的训练效果. 而且上述工作仅关注分类难易不均衡

性, 并未关注正负例样本数量不均衡性.

(3) 定位质量不均衡, 即相比于负例样本, 正例样本的初始定位质量较差. 在单阶段目标检测算法的训练阶段, 仅当锚点框与真实边界框的交并比(intersection over union, IOU)达到一定阈值(通常为 0.50), 该锚点框才会被选为正例样本以进行物体位置和尺寸回归, 余下的锚点框则作为负例样本. 如图 2“初始锚点框”所示, 本文发现用于回归的正例样本与真实边界框的 IOU 呈现出如下特点: 大部分正例锚点框的 IOU 在 0.50 附近; 但随着IOU值的增加, 正例锚点框数量迅速减少. 这说明正例样本的初始定位质量较差, 会导致预测的边界框定位质量也较差, 进而降低了检测精度[15].

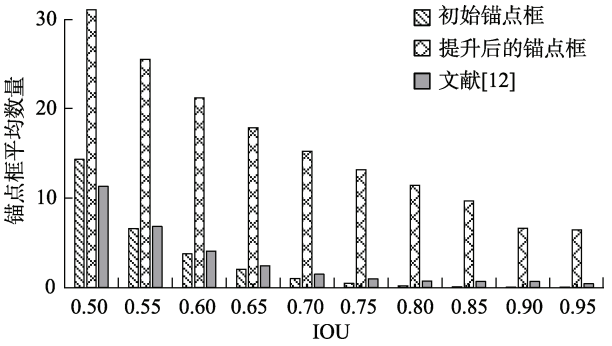

图 2　不同 IOU 下的锚点框平均数量

尽管两阶段算法在生成区域建议框时会调整初始设置的锚点框位置和尺寸; 单阶段算法中, 文献[11-12]也对初始锚点框进行调整. 这些工作利用 Objectness 直接筛选调整后的锚点框, 未考虑调整后锚点框与真实边界框的 IOU, 且仅采样部分筛选后的锚点框训练检测网络, 因此锚点框的定位质量难以保证(如图 2“文献[12]”所示). 文献[14]提出增加位置回归损失中简单样本(即与真实边界框 IOU 较大的正例样本)的权重, 同时降低难样本(即与真实边界框 IOU 相对较小的正例样本)的损失权重来改善预测的边界框的定位质量, 但未解决正例样本初始定位质量的问题.

因此, 本文提出识别并抑制简单负例锚点框, 同时提高正例锚点框的数量和定位质量的方案, 从而缓解基于锚点框的单阶段目标检测算法的不均衡问题.

2　本文算法

针对单阶段目标检测算法的正负锚点框不均衡问题, 本文提出 APM, 旨在为训练检测网络提供均衡的锚点框. 本文还提出 FAM, 根据锚点框的变化, 动态提取对齐的特征表达, 以利于检测网络的训练及预测.

2.1　APM

本文所提 APM 结构简洁. 对于各锚点框而言, APM 使用一个分类器(APM_C)来预测该锚点框是正例锚点框的可能性大小; 用一个回归器(APM_ R)粗略地调整锚点框的初始位置和尺寸. 经过 APM 的处理之后, 用于训练检测网络的锚点框除了具有位置和尺寸等几何信息之外, 还具有了新的属性: 是正例锚点框的可能性.

为了改善检测网络训练时的不均衡问题, 对 APM 输出的锚点框采用如下均衡策略: (1) 利用锚点框几何信息与真实边界框匹配之后为负例; (2) 锚点框是正例锚点框的可能性小于某一阈值 θ (本文设置 θ=0.01); (3) 检测网络的分类器在训练时识别满足策略(1)(2)的锚点框并忽略其梯度. 此时检测网络将更多地关注所有的正例锚点框和难负锚点框.

结合以上均衡策略, 下面定性地分析 APM 输出的锚点框对不均衡的影响, 将在实验部分定量地分析其对检测精度的影响:

(1) 如果 APM 只有 APM_C, 即只输出锚点框是正例的可能性. 经均衡策略处理后, 用于训练的正例锚点框与负例锚点框的数量比从 28.4∶18 198.3 增加至 28.4∶1 911.6 (如图 3“APM_C”所示), 而正例锚点框的数量和定位质量并没有改变, 该配置仅能部分改善数量不均衡.

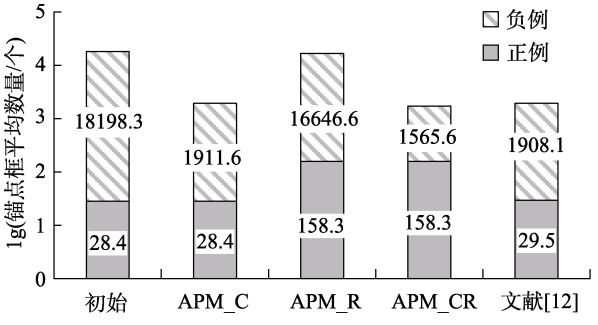

图 3　不同策略下正负例锚点框平均数量

(2) 如果 APM 只用 APM_R, 即只粗略地调整锚点框位置和尺寸, 如图 3“APM_R”所示. 正例锚点框的数量明显地增加, 此时正例锚点框数量与负例锚点框的数量比值从 1∶625 增加至 1∶105 可以较为显著地改善数量不均衡问题. 而由于负例锚

点框数量依然多于正例锚点框数量，只是部分缓解数量不均衡. 在各 IOU 值下的正例锚点框平均数量也明显地增加，如图 2“提升后的锚点框”所示，说明 APM_R 不仅能提升正例锚点框的数量，也能很好地改善正例锚点框的定位质量. 但是，此时无法明确简单负例锚点框(仅能利用均衡策略(1))，因而不能缓解分类难易不均衡.

(3) APM 同时配置 APM_C 和 APM_R，如图 3“APM_CR”所示. 结合均衡策略，训练时的负例锚点框数量减少而正例锚点框数量增加，相比于初始设置时，正例锚点框和负例锚点框数量比值增加了 65 倍，接近 1:10. 该配置下能够很好地缓解数量不均衡. 如图 2“提升后的锚点框”所示，正例锚点框的定位质量明显改善，也缓解了定位质量不均衡问题. 该配置下的具体结构如图 4 右上角所示.

对比两阶段检测算法[3-4]及相关单阶段检测算法[12-13]，它们直接利用 Objectness 分数过滤大部分的锚点框，不考虑与真实边界框位置关系，再筛选固定数量满足一定比值的正负样本训练检测网络. 其仅缓解了样本子集内的数量均衡，而未有效缓解定位质量不均衡，且仅有部分样本用于训练，检测网络无法提取出更多鲁棒性的特征表达. 而本文中，APM 输出的锚点框全部用于训练检测网络，在计算损失及梯度时抑制负例锚点框的影响，因此检测网络能从全部锚点框的特征表达中挖掘出判别性更强的特征. 而 APM 提升的全部正例锚点框及未忽略梯度的负例锚点框，有效地缓解数量及定位质量不均衡.

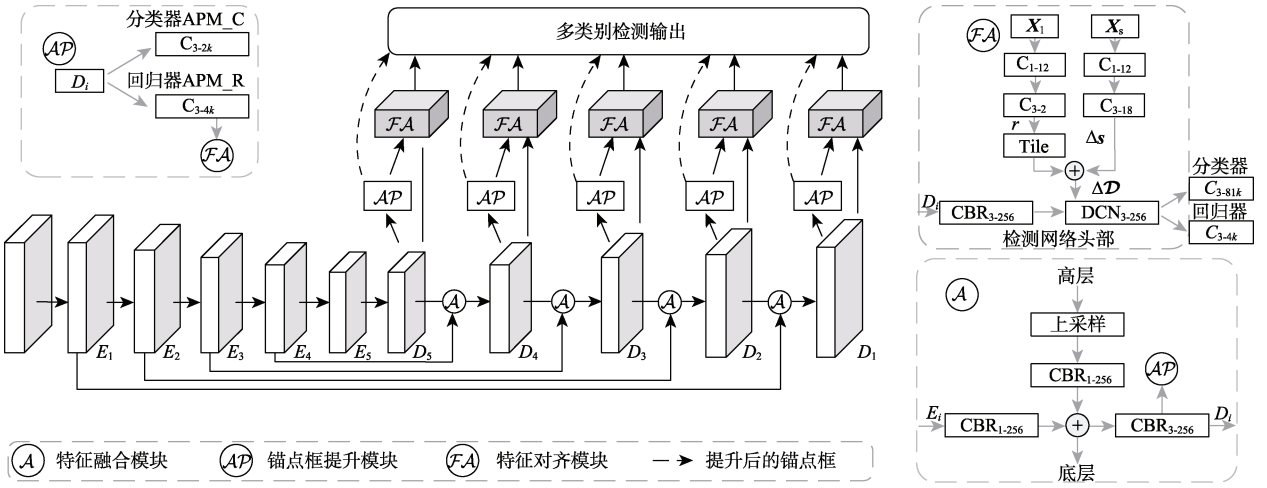

图 4 网络整体结构图

对于分类难易不均衡，文献[16]从损失值角度筛选出损失值最高的样本作为难样本重点训练；文献[13-14]并未明确定义难易样本而调节所有样本的损失权重. 本文从概率(APM_C 和均衡策略(2))及位置(APM_R 和均衡策略(1))2 个方面定义分类难易样本，并在训练分类器时忽略简单负例样本的梯度，以此减少简单负样本的梯度对网络训练的影响，而使检测网络更多地关注难负样本及正例样本.

另外，本文提出的 APM 类似于文献[4]中的区域建议网络(region proposal network, RPN)，但是本文所提模块与 RPN 存在如下不同: (1) RPN 输出的边界框会经过过滤低质量边界框、非极大值抑制等操作；而 APM 主要是为了改善初始设置的锚点框，因此不需要上述操作. (2) RPN 最终输出的建议框中只有前 n (通常 n =2 000)个建议框用于后续的检测. 相反 APM 的输出锚点框会全部输入至检测网络中. 尽管本文将 APM 输出的正例锚点框可能性用于检测网络，但是该属性仅仅用做指示器，即识别简单负例样本及指示检测网络的分类器在训练时忽略简单负例的梯度，而不像 RPN 用于预先过滤锚点框. (3) RPN 在输出稀疏的建议框之后与后续的检测网络之间没有交互；相反，本文提出 APM 的分类器输出结果用于指示检测网络的训练，同时该模块的回归器输出有助于检测网络提取与提升后的锚点框对齐的特征，用于训练和预测. (4) RPN 输出的稀疏建议框采用 ROIPooling[3]或 ROIAlign[17]等计算和存储资源消耗较大的算法提取建议框内部的特征，用于后续检测网络训练和预测；而本文提出了一个能够高效地为密集输入的锚点框提取对齐特征的模块.

2.2　FAM

在基于锚点框的单阶段目标检测中, 预先设置的锚点框的中心点与特征图的每个位置相对应, 因此可以使用锚点框中心点所在位置的特征进行预测[9]. 但是经过 APM 的处理之后, 预先设置的锚点框的位置和尺寸发生改变, 因此不能简单地使用原有的特征进行训练和预测. 为此本文提出一个 FAM 以提取与调整后的锚点框对齐的特征表达.

为了获得对齐的特征表达, 一种可能的方式是: 在新的锚点框位置采样特征, 然后缩放至固定大小, 最后使用全连接或者卷积层提取特征. 但是, 这种操作经历了一个复杂的流程(采样 $\rightarrow$ 缩放 $\rightarrow$ 卷积/全连接等), 其类似于两阶段网络采用的 ROIPooling 或 ROIAlign 算法, 在输入密集数量的锚点框时将消耗较多的存储和计算资源. 受可变形卷积操作[6]启发, 本文提出一种通过偏移卷积核同时采样与卷积提取特征的 FAM. 该模块的关键操作在于: 如何能够随着 APM 输出的锚点框的位置和尺寸变化, 自动地获取合适的采样卷积核的偏移量(表示为 $\Delta\boldsymbol{\mathcal{D}}$). 如图 5 所示, 有 3 种不同的偏移量获取方式.

(1) 隐式学习. 一种简单地做法是, 类似于文献[6]中的可变形卷积操作, 如图 5a 所示, 从上一层的特征图中隐式学习偏移量, 即 $\Delta\boldsymbol{\mathcal{D}}=\mathcal{F}(\boldsymbol{X})$. 其中, $\boldsymbol{X}$ 表示上一层的特征; $\mathcal{F}$ 表示偏移量学习函数. 这种学习方式具有一定的局限性: 偏移量的学习与锚点框的位置和尺寸变化量无关, 只由目标损失函数隐式地优化. 但此时网络提取的特征并未与提升后的锚点框对齐, 因此目标损失函数只能提供次优化的反向梯度信息, 进而导致有限的性能改善.

(2) 显式学习. 由于 APM 的回归器能输出锚点框位置和尺寸的变化信息, 因此该信息可以指导偏移量的学习. 如果仅采用图 5b 所示的 APM 回归器输出的位置变化信息作为输入, 即 $\Delta\boldsymbol{\mathcal{D}}=\mathcal{F}(\boldsymbol{X}_{\mathrm{l}})$. 其中 $\boldsymbol{X}_{\mathrm{l}}$ 表示位置变化信息. 此时由于缺少锚点框的尺寸变化信息, 因此得到的偏移量 $\Delta\boldsymbol{\mathcal{D}}$ 并不是最优的; 而只采用图 5b 所示的 APM 回归器输出的尺寸变化信息作为输入, 即 $\Delta\boldsymbol{\mathcal{D}}=\mathcal{F}(\boldsymbol{X}_{\mathrm{s}})$. 其中 $\boldsymbol{X}_{\mathrm{s}}$ 表示尺寸变化信息. 此时由于缺少锚点框的位置变化信息, 因此得到的偏移量 $\Delta\boldsymbol{\mathcal{D}}$ 并不能定位到合适的采样位置. 如果采用图 5c 所示的将位置变化信息 $\boldsymbol{X}_{\mathrm{l}}$ 和大小变化信息 $\boldsymbol{X}_{\mathrm{s}}$ 拼接作为输入, 即 $\Delta\boldsymbol{\mathcal{D}}=\mathcal{F}(\boldsymbol{X}_{\mathrm{l}};\boldsymbol{X}_{\mathrm{s}})$. 由于 $\boldsymbol{X}_{\mathrm{l}}$ 和 $\boldsymbol{X}_{\mathrm{s}}$ 位于不同的特征空间, 直接将这 2 种不同尺度的特征耦合在一起, 并不能充分地提取出最优的特征信息用于优化.

(3) 解耦式学习. 锚点框位置变化信息 $\boldsymbol{X}_{\mathrm{l}}$ 和大小变化信息 $\boldsymbol{X}_{\mathrm{s}}$ 位于不同尺度特征空间, 不能直接耦合. 因此, 本文提出图 5d 所示的解耦式学习方式, 将偏移量进行分解学习, 即 $\Delta\boldsymbol{\mathcal{D}}=r+\Delta\boldsymbol{s}=\mathcal{F}(\boldsymbol{X}_{\mathrm{l}})+\mathcal{G}(\boldsymbol{X}_{\mathrm{s}})$. 其中, r 是标量, 表示从 $\boldsymbol{X}_{\mathrm{l}}$ 学习的卷积核总的位置变化; $\Delta\boldsymbol{s}$ 表示从 $\boldsymbol{X}_{\mathrm{s}}$ 中学习的卷积核中每个位置的偏移残差量; $\mathcal{G}$ 表示残差学习函数. 这种解耦式的学习方式的直观解释是: 位置变化信息 $\boldsymbol{X}_{\mathrm{l}}$ 有助于卷积核首先找到最优的整体采样位置, 而锚点框尺寸变化信息 $\boldsymbol{X}_{\mathrm{s}}$ 进一步优化卷积核的每个位置以找到特征图上最优的采样位置.

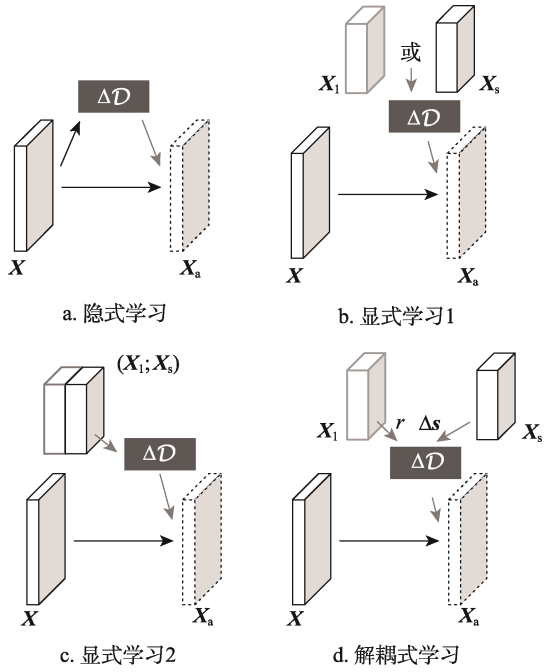

图 5　采样卷积核偏移量不同获取方式

表 1 所示为 3 种偏移量获取方式的检测精度. 基准结果是没有采用 FAM 的检测精度(表 1 第 1 行 "无对齐"). 从表 1 中可以看出, 本文提出并采用的解耦式学习方式能够获得最好的检测精度. 图 4 右

表 1　采样卷积核偏移量不同获取方式时的检测精度　%

学习方式		AP	AP_50	AP_75	AP_s	AP_m	AP_l
无对齐		31.7	53.1	33.1	16.0	35.4	45.4
隐式学习		32.8	54.0	34.5	16.8	36.3	46.7
显式学习	$F(\boldsymbol{X}_{\mathrm{l}})$	33.6	55.0	35.7	18.2	37.4	47.4
	$F(\boldsymbol{X}_{\mathrm{s}})$	33.4	54.6	35.0	17.6	36.8	47.8
	$F(\boldsymbol{X}_{\mathrm{l}};\boldsymbol{X}_{\mathrm{s}})$	34.1	55.5	36.1	17.9	37.9	47.9
解耦式学习		34.8	55.4	37.5	18.6	38.2	49.2

上角展示了采用该学习方式时的网络结构，该结构输出的 $\Delta \boldsymbol{s}$ 有 $2K$（K 表示卷积核大小，本文中 K=9，即 3×3 的采样卷积核)个通道，r 有 2 个通道，2 个输出分支按元素相加作为卷积核的偏移量 $\Delta \boldsymbol{\mathcal{D}}$．将该偏移量施加到采样卷积核上，即是能同时采样和卷积的 FAM.

除了表 1 定量地对比 3 种不同学习方式外，图 6 定性地对比了采用不同学习方式时的采样位置(其中玫红色矩形框、绿色矩形框和蓝色矩形框分别表示初始锚点框、提升后的锚点框和真实边界框；玫红色点表示标准卷积的采样位置．图 6d 中红色点表示卷积核加上 r 后的采样位置，而绿色点表示最终的采样位置).

图 6 表明，采用隐式学习方式时，只在标准卷积采样位置附近提取特征，并且提取到较多的无关信息，尤其当物体尺寸较小的时候(如图 6 中的黑色轿车)．在显式学习方式中，APM 的回归器输出的位置变化信息和尺寸变化信息确实有助于采样到合适的特征，但是位置变化信息和尺寸变化信息的耦合也导致了次优化的偏移量．而解耦式学习首先借助 $\boldsymbol{X}_{\mathrm{l}}$ 定位到合适的位置(图 6d 中的红色的点)，然后 $\boldsymbol{X}_{\mathrm{s}}$ 进一步精调卷积核到最优的采样位置．因此解耦式学习方式能够根据锚点框调整之后的位置和尺寸变化，找到特征图上最合适的位置．本文所提的 FAM 即采用该学习方式提取对齐的特征，用于训练和预测．

a. 输入 b. 隐式学习 c. 显式学习 d. 解耦式学习

图 6 不同学习方式时的采样位置

3 实验设置

3.1 网络结构

编解码结构的网络因其能够较好地融合底层细节信息和高层语义信息，从而在语义分割、目标检测[5,17]中广泛应用．因此本文采用该形式的主干网络结构．类似于单发多框检测算法(single-shot multibox detector, SSD)[9]，本文使用 VGG-16[18]作为主干特征提取网络．将 VGG-16 网络中的 conv4_3, conv7, conv8_2, conv9_2, conv10_2 分别记作 $\{E_1, E_2, E_3, E_4, E_5\}$，它们对应的步长分别为 s={8, 16, 32, 64, 128}．解码结构起始于 E_5，并使用 256 通道的 1×1 卷积、BatchNorm[19]和 ReLU 级联的结构(记作 $\mathrm{CBR}_{x\text{-}y}$，$x$ 表示卷积核大小为 $x\times x$，y 表示卷积输出通道数)提取特征．解码结构使用双线性插值逐渐上采样特征图，每个上采样之后的特征图进一步使用 $\mathrm{CBR}_{3\text{-}256}$ 以增强上采样后的特征表达能力．如图 4 所示，解码结构每层分别记作 $\{D_1, D_2, D_3, D_4, D_5\}$，其分别与 $\{E_1, E_2, E_3, E_4, E_5\}$具有相同的特征图大小，本文还通过下采样 D_5 得到新的解码层 D_6．

与 RetinaNet[13]采用分类与回归分支分离且各自有多层结构不同，本文采用一个简洁的检测网络头部，即一个 $\mathrm{CBR}_{3\text{-}256}$，其后连接分类层和回归层，如图 4 右上角所示．本文采用大小为 384×384 和 512×512 的图片作为输入．图 4 中的 k 表示锚点框数量．

3.2 锚点框的设置与匹配

类似于 SSD[9]，本文选择 $\{D_1, D_2, D_3, D_4, D_5, D_6\}$层来生成锚点框．具体地，在特征图 $\{D_1, D_2, D_3, D_4\}$的每个位置设置 1 种尺度、6 种长宽比(即 1, 1, 2, 3, 1/2, 1/3)的锚点框，在 $\{D_5, D_6\}$设置 1 种尺度、4 种长宽比(即 1, 1, 2, 1/2)的锚点框．

本文采用双向匹配原则以匹配锚点框和真实边界框，即：(1) 每个真实边界框匹配与其 IOU 最大的锚点框；(2) 与任一真实边界框的最大 IOU 大于 0.50 的锚点框作为正例，并以该真实边界框作为训练目标，与任意真实边界框之间的 IOU 都小于 0.30 的作为负例．

3.3 网络训练设置

本文所提网络的训练损失函数由 2 个部分组成：APM 的损失函数和检测网络的损失函数．该损失函数为

$$L(p, x, c, t) = (1/N_{\mathrm{APM}})(L_{\mathrm{b}}(p, [y \geqslant 1]) + [y \geqslant 1]L_{\mathrm{r}}(x, g)) + (1/N_{\mathrm{d}})(L_{\mathrm{cls}}(c, y) + [y \geqslant 1]L_{\mathrm{r}}(t, g)).$$

其中，N_{APM} 和 N_{d} 分别表示 APM 和检测网络训练

时的正例样本数量; p 表示是正例锚点框的可能性大小; y 是真实类别标签; g 表示真实边界框; L_b 是二元交叉熵损失函数; L_{cls} 是 Softmax 损失函数, 用于预测具体的类别; L_r 是 Smooth L_1 损失函数[3], 用于调整锚点框位置和大小.

主干网络 VGG-16 预先使用 ImageNet 训练, 然后在目标检测数据集上利用动量为 0.9、权重衰减率为 0.0005 的随机梯度下降法精调. 新增加的网络层使用 MSRA 初始化. 训练时的批量大小为 32, 并采用 SSD 中所述的数据增强方法, 即水平翻转、缩放和光照变化等.

3.4　网络测试设置

在测试阶段, 锚点框经 APM 处理后全部输入检测网络以预测边界框及具体类别. 如果 APM 输出的某个锚点框是正例锚点框的可能性小于 θ, 其相应的预测边界框便被忽略. 余下的预测边界框经过非极大值抑制之后, 取前 300 个作为每张图片的检测结果用于评测.

4　实验结果及分析

为验证本文算法及检测网络的有效性, 使用公开的目标检测数据集 MS COCO 进行实验. 在训练时使用常用的共有 11 万张图片的 trainval35k 作为训练集, 使用包含 5000 张图片的 minival 作为验证集, 最后使用测试集 test-dev 进行测试, 并将预测结果提交至在线评估服务器, 得到最终的检测精度. 训练初始学习率为 2×10^{-3}, 在 100 和 140 个训练周期之后将学习率分别下降为 2×10^{-4} 和 2×10^{-5}, 总的训练周期为 160 个.

4.1　所提模块的有效性

(1) APM. 上文已定性地验证该 APM 输出的锚点框对改善不均衡问题的效果. 在此定量地验证其对检测精度的影响, 如表 2 所示.

表 2　所提模块的有效性　　%

APM_C	APM_R	FAM	mAP		
			AP	AP_50	AP_75
			29.6	49.4	30.8
✓			31.0	51.3	32.6
	✓		30.6	50.5	32.4
✓	✓		31.7	53.1	33.1
✓	✓	✓	34.8	55.4	37.5

如果 APM 只有分类器(APM_C), 即只预测每个锚点框是正例的可能性, 训练时检测网络识别出可能性小于 θ 的简单负例锚点框并忽略其梯度, 此时平均精度均值(mean average precision, mAP)从 29.6%提升至 31.0%. 这说明 APM_C 虽仅部分缓解数量不均衡, 但能改善检测精度. 如果 APM 只有回归器(APM_R), 即只改善初始设置的锚点框的位置和尺寸, 此时正例锚点框的数量和定位质量均被提升; 但是负例锚点框的数量依然较多, 仍会对检测网络的训练产生影响, 这种配置只带来有限的检测精度改善. 而 APM 同时配置分类器和回归器时(APM_CR), 不仅可以大大地改善不均衡现象, 而且 mAP 能够提升 2.1%(从 29.6%到 31.7%).

(2) FAM. 表 2 中的结果说明本文所提的 FAM 使 mAP 进一步提升 3.1%(从 31.7%到 34.8%). 尽管 APM 输出的锚点框有助于改善不均衡问题, 但是为每个提升后的锚点框提取到适合用于预测的特征表达, 对检测精度的改善也至关重要. 检测精度的明显改善也证明本文的解耦式学习方式对提取到对齐的特征表达的合理性.

(3) 定位精度. APM 除了有助于解决单阶段检测网络的不均衡问题外, 还能改善最终预测边界框的定位精度. 如表 3 所示, 所提的 APM 和 FAM 一致地改进了不同 IOU 情况下的检测精度. 从表 3 中可以看出, 高 IOU 值下的检测精度改进不及低 IOU 时的改进, 例如在 IOU=0.70 时 mAP 提升了 5.4%, 而在 IOU=0.85 时 mAP 只提升了 3.2%. 导致这种现象的原因可能是: 在训练检测网络时采用了较低的正例样本的匹配条件(IOU≥0.50), 因此检测网络在高 IOU 条件下只能获得次优化的检测结果[15]. 一种可能的解决方案是级联多个检测网络以及 FAM, 并采用逐渐增加的正例匹配 IOU 来训练检测网络[15], 这个解决方案将作为未来工作进一步讨论.

表 3　物体在不同 IOU 下的检测精度　　%

APM_R	FAM	AP_50	AP_60	AP_70	AP_80	AP_85
		49.4	44.2	36.4	24.1	16.0
✓		53.1	47.9	39.1	25.3	16.5
✓	✓	54.7	49.7	41.8	28.6	19.2

(4) 不同尺寸的检测精度. 表 4 显示了不同尺寸的物体在不同 IOU 值下的检测精度. 从表中可以得到如下结论: a. 对于中等尺寸和大尺寸的物体, 它们检测精度的改进量在 IOU 范围为 0.50~0.80 时随着 IOU 值的增加而增加; b. 小尺寸的物体的总体检测精度获得了很大改进, mAP 从

12.1%提高至 18.6%; c. 小尺寸物体的检测精度改进量随着 IOU 值的增加逐渐减少, 这是因为只有大约 34.3%的小尺寸物体有超过 2×2 大小的特征存在用于检测的最大分辨率的特征图上(E_1), 因此大多数的小尺寸物体无法提取到有用的判别性特征信息.

表 4 不同尺寸的物体在不同 IOU 下的检测精度 %

算法	尺寸	AP	AP_50	AP_60	AP_70	AP_80	AP_85
基准	小	12.1	24.6	20.1	14.2	6.9	3.8
	中	32.4	56.3	50.0	40.5	24.4	14.7
	大	44.3	67.7	62.4	54.1	41.3	30.0
本文	小	18.6	35.1	29.8	22.9	12.4	6.6
	中	38.6	62.6	57.1	48.9	32.5	19.9
	大	49.2	70.7	66.8	60.7	48.1	35.8

(5) 与 Focal loss 对比. Focal loss 解决不均衡的方式是减少已被很好分类的锚点框的损失及梯度权重; 而本文采用的策略是利用 APM 的输出从概率和位置角度识别并忽略易分类的负例锚点框的梯度, 同时提高正例锚点框的数量和定位质量. 因此, 为了与 Focal loss 公平对比, 本文采用 2 种实验方案: a. 使用本文采用的尺寸 384×384 作为输入, APM 只保留回归器, 检测网络的分类损失函数使用 Focal loss (表示为 RetinaNet_AFF). b. 采用 RetinaNet 的输入尺寸, 即最短边为 800 像素, 并将本文提出的 APM 和 FAM 以及解决不均衡的策略应用到 RetinaNet.

实验结果如表 5 所示, 在这 2 种实验方案中, 本文算法总能获得比采用 Focal loss 更高的检测精度. 这可能是因为 Focal loss 仅仅减少了已被很好分类的锚点框的损失和梯度权重, 但是这些梯度仍然存在于训练过程中, 大量的负例锚点框的梯度仍然会影响检测网络的训练. 而本文直接忽略了易分类负例锚点框的梯度, 使得网络不会过多关注容易分类的特征信息, 而挖掘和提取判别性及泛化性更强的特征信息用于识别难负样本和正例样本.

表 5 与 Focal loss 的检测精度对比

主干网络	输入尺寸	算法	mAP/%
ResNet-101	384×384	RetinaNet[13]	32.1
		RetinaNet_AFF	34.6
		本文	36.9
ResNet-50	1333×800	RetinaNet[13]	35.7
		RetinaNet_AFF	37.6
		本文	39.6

(6) 与其他特征对齐算法的对比. 文献[20]也提出与本文类似的特征对齐算法, 但是不同于本文特征对齐算法采用学习的方式先定位再精调到最优采样位置的策略, 文献[20]采用手动设置卷积核的偏移量. 为了与其进行公平对比, 本文采用与文献[20]相同的实验设置, 实验结果如表 6 所示.

表 6 与其他特征对齐算法检测精度对比 %

算法	AP	AP_50	AP_75	AP_s	AP_m	AP_l
文献[20]	35.6	55.9	38.3	19.5	38.5	47.2
本文	36.8	56.0	40.0	21.4	40.0	49.5

实验结果表明, 本文所提 FAM 能够获得更好的实验性能. 这是因为如图 6d 所示, 最优的卷积位置可以在特征图上任意提供判别性特征的位置. 文献[20]手动设置的采样点偏移量并限制在锚点框内部, 导致提取不到判别性更强的特征信息, 进而只能获得次优化的检测结果, 而文献[20]为了改进检测精度使用了比本文更深更复杂的检测头部网络以及使用更高 IOU 值进行匹配[15]. 这也进一步证明了本文所提 FAM 的简洁性、合理性和有效性.

4.2 与现有算法对比

4.2.1 检测精度对比

为了与现有算法进行公平对比, 本文采用 384×384, 512×512 和最短边为 800 像素的图片作为输入, 使用 VGG-16 和 ResNet-101[21]作为特征提取网络, 测试集为 test-dev, 并将检测结果提交到评估服务器以获得检测精度, 如表 7 所示, 其中“多尺度”表示多尺度训练. 以 512×512 的图片作为输入并采用 VGG-16 网络时, 本文算法 mAP=37.6%, 甚至超过了采用 ResNet-101 的以大尺寸作为输入的 RetinaNet 800. 当本文采用 ResNet-101 时能够获得更高的检测精度, 超过了现有的大多数检测算法. 当采用大尺寸 800 像素作为输入时 mAP=41.4%, 而结合多尺度输入训练时获得了 mAP=42.8%的最好检测精度.

4.2.2 测试时间对比

表 7 中同时与现有文献的运行速度进行了对比. 本文的测试环境为 NVIDIA GTX 1080 Ti, PyTorch 0.4.1, i7-6850k CPU, CUDA9.0 and cuDNN v7. 从表 7 第 3 列可以看出: (1) 在相同检测精度下, 本文算法获得了更快的测试速度; (2) 相同测试速度下, 本文算法获得了更好的检测精度.

图 7 更直观地展示了本文算法的测试时间相比于其他算法的优势. 本文算法获得较短的测试时间的原因在于: (1) 本文算法的检测网络头部只

有一层卷积层, 而不像 RetinaNet 采用分类和回归分离且多层的检测头部; (2) 简洁而合理的 FAM 比两阶段网络所采用的 ROIPooling 或 ROIAlign 算法更高效; (3) 小尺寸的输入消耗较短的测试时间.

表 7　现有检测算法性能对比

算法	主干网络	输入尺寸	速度/(帧·s^{-1})	检测精度/%					
				AP	AP_50	AP_75	AP_s	AP_m	AP_l
Faster R-CNN[4]	VGG16	1 000×600	7.0	21.9	42.7				
R-FCN[22]	ResNet101	1 000×600	9.0	29.9	51.9		10.8	32.8	45.0
FPN[5]	ResNet101	1 333×800	6.0	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN[17]	ResNet101	1 333×800	5.1	38.2	60.3	41.7	20.1	41.1	50.2
Cascade R-CNN[15]	ResNet101	1 333×800	7.1	42.8	62.1	46.3	23.7	45.5	55.2
DCN v2[23]	ResNet101	1 333×800		44.0	65.9	48.1	23.2	47.7	59.6
SSD[9]	ResNet101	513×513		31.2	50.4	33.3	10.2	34.5	49.8
DSSD[24]	ResNet101	513×513	5.5	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet 800[13]	ResNet101	1 333×800	5.1	37.8	57.5	40.8	20.2	41.1	49.2
RetinaNet 800[13](多尺度)	ResNet101	1333×800	5.1	39.1	59.1	42.3	21.8	42.7	50.2
RefineDet[11]	VGG16	512×512	24.1	33.0	54.5	35.5	16.3	36.3	44.3
RFBNet[25]	VGG16	512×512	30.3	34.4	55.7	36.4	17.6	37.0	47.6
PFPNet-R[26]	VGG16	512×512	22.2	35.2	57.6	37.9	18.7	38.6	45.9
CornerNet[27]	Hourglass 104	511×511	4.1	40.5	56.5	43.1	19.4	42.7	53.9
GA-RetinaNet[28]	ResNet101	1 333×800		37.1	56.9	40.0	20.1	40.1	48.0
FSAF[29]	ResNet101	1 333×800	5.6	40.9	61.5	44.0	24.0	44.2	51.3
RPDet[30]	ResNet101	1 333×800	9.5	41.0	62.9	44.3	23.6	44.1	51.7
FCOS[31]	ResNet101	1 333×800	13.0	41.5	60.7	45.0	24.4	44.8	51.6
AlignDet[20]	ResNet101	1 333×800	9.1	42.0	62.4	46.5	24.6	44.8	53.3
本文 V384	VGG16	384×384	62.5	35.2	55.9	38.1	17.7	38.2	48.3
本文 V512	VGG16	512×512	38.5	37.6	58.7	41.0	21.0	40.4	49.5
本文 R384	ResNet101	384×384	40.0	36.9	57.0	40.2	16.4	40.4	53.3
本文 R512	ResNet101	512×512	28.6	40.0	60.8	43.8	21.3	43.9	53.9
本文 R800	ResNet101	1 333×800	12.5	41.4	60.4	45.5	22.7	44.7	53.2
本文 R800(多尺度)	ResNet101	1 333×800	12.5	42.8	62.2	47.0	24.3	46.2	53.8

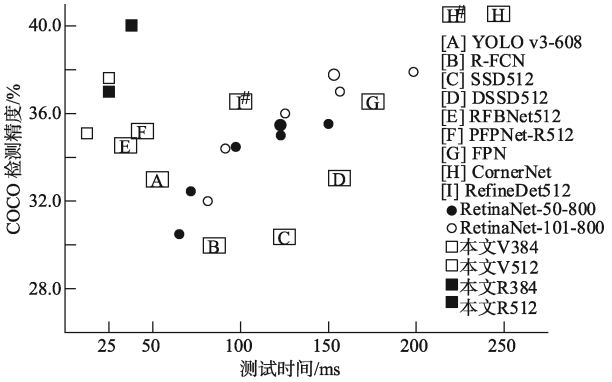

图 7　与现有算法测试时间对比

4.2.3　与两阶段目标检测算法的对比

尽管本文已定性地叙述与两阶段目标检测算法中的 RPN 的区别以及解决数量和定位质量不均衡的异同. 为了更直观地展示与两阶段目标检测算法不同, 表 8 展示了两阶段网络 Faster R-CNN 与本文算法在运行时经历的不同阶段及每个阶段的耗时和所使用锚点框的数量差异.

不同于两阶段算法的 RPN 输出需要经历一次标准的检测流程(即过滤低质量输出、非极大值抑制和选择固定量的结果)且消耗较多时间, ROI-Pooling 或 ROIAlign 算法处理稀疏的建议框且其处理时间随着区域建议框数量的增加而逐渐增加. 本文所提的 APM 的输出直接作为检测网络的输入, 即密集的锚点框输入(单阶段检测网络特色, 这也是本文算法称之为单阶段网络的原因), 而所提 FAM 能够高效地提取密集输入的锚点框所需的特征表达. 即使在大尺寸输入导致更密集的锚点框时, 本文算法相比于两阶段算法仍然可以更快速地运行.

另外, 表 8 中也表明, 本文算法与两阶段算法解决不均衡问题的差异: 两阶段算法使用 RPN 生

表 8 本文算法与两阶段网络算法各处理阶段及耗时对比

算法	主干网络	输入尺寸	主干网络时间/ms	锚点框处理时间/ms		锚点框后处理时间(NMS)/ms	特征对齐方式时间/ms		检测子网络时间/ms	检测结果后处理时间/ms
				RPN	APM		ROIAlign	FAM		
Faster R-CNN[4]	ResNet101	384×384	17.2	0.6		8.0/5.0	1.9/1.3		0.4/0.3	13.4/5.4
Faster R-CNN[4]	ResNet50	1 333×800	16.6	1.6		45.4/35.0	8.4/3.1		0.4/0.3	14.2/5.7
本文 384	ResNet101	384×384	17.4		1.0			1.5	0.1	5.1
本文 800	ResNet50	1 333×800	16.9		1.5			2.8	0.4	13.2

注. ·/·表示建议框数量分别为 1 000/300 个时的耗时.

成区域建议框之后, 在训练和测试阶段会利用 RPN 输出的分类分数过滤掉大部分输出, 仅保留少量区域建议框(如 1 000 个或 300 个), 从中采样定量的区域建议框, 并设置正负样本为固定比值(比如 1∶3); 因此, 其解决不均衡问题是相对的, 即在样本子集范围内而非全部样本内. 该算法在样本输入时解决不均衡问题, 而本文使用 APM 的全部输出, 即解决全部样本的不均衡问题. 此外, 本文是在计算损失及梯度时解决不均衡问题, 因而可以使网络在训练时挖掘出更多判别性特征.

4.2.4 PASCAL VOC 结果对比

为更好地验证本文算法的有效性, 在 PASCAL VOC 2007 数据集上进行实验. 使用 PASCAL VOC 2007 与 PASCAL VOC 2012 的训练验证图片的合集作为训练集, 在 PASCAL VOC 2007 测试集上进行测试. 初始学习率设为 4×10^{-3}, 学习率在 150 和 200 个训练周期后依次降为 4×10^{-4} 和 4×10^{-5}, 总的训练周期为 250 个.

实验结果如表 9 所示, 从中可以看出, 当输入尺寸为 384×384 时, 本文算法 mAP=82.0%, 超过了现有的一些使用大尺寸输入的单阶段及两阶段目标检测算法. 而输入尺寸为 512×512 时获得了更好的检测精度, 相比于文献[11,25-26]中的单阶段算法, 通过增加网络层数[11]、使用复杂的包含大孔率的带孔卷积的结构[25]及并行的特征金字塔特征融合方式[26], 本文仅采用简洁的结构与简单的不均衡缓解策略, 对资源和时间的消耗也相对较少. 这也证明了本文算法及检测网络的有效性.

表 9 PASCAL VOC 上检测精度对比

算法	主干网络	输入尺寸	mAP/%
Faster R-CNN[4]	ResNet101	1 000×600	76.4
R-FCN[22]	ResNet101	1 000×600	80.5
DCR[32]	ResNet101	1 000×600	82.5
SSD[9]	VGG16	512×512	79.8
RefineDet[11]	VGG16	512×512	81.8
RFBNet[25]	VGG16	512×512	82.2
PFPNet-R[26]	VGG16	512×512	82.3
本文 384	VGG16	384×384	82.0
本文 512	VGG16	512×512	82.7

5 结 语

单阶段目标检测算法能获得实时的运行速度, 因而获得了较多的关注与应用, 但是训练时的不均衡问题影响了该算法的检测精度. 本文首先具体地分析了 3 种不均衡性, 即数量不均衡、分类难易不均衡和定位质量不均衡, 及现有文献对这 3 种不均衡性的解决方案的优缺点. 进而提出在训练时抑制负例锚点框的数量, 同时提升正例锚点框的数量和定位质量的策略. 为此本文设计了 APM 预测预先设置的锚点框是正例锚点框的可能性, 并调整锚点框位置和尺寸以提升正例锚点框的数量和定位质量. 为了能够提取与提升后的锚点框对齐的特征, 以更有效地训练检测网络, 本文设计了简洁高效的以解耦方式学习获取采样位置的 FAM. 在训练检测网络时根据锚点框是正例的可能性, 分类器从概率和位置角度识别简单易分类锚点框并忽略其梯度. 在公开的目标检测数据集上的实验结果表明, 本文算法能够显著改进单阶段目标检测算法的检测精度, 同时能够维持实时的运行速度. 在未来工作中, 考虑将该算法进行压缩或者精简以在计算资源有限的嵌入式和移动设备上应用该单阶段目标检测算法.

参考文献(References):

[1] Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2014: 580-587

[2] He K M, Zhang X Y, Ren S Q, *et al*. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904-1916

[3] Girshick R. Fast R-CNN[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2015: 1440-1448

[4] Ren S Q, He K M, Girshick R, *et al*. Faster R-CNN: towards real-time object detection with region proposal networks[C] //Proceedings of the Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2015: 91-99

[5] Lin T Y, Dollár P, Girshick R, *et al*. Feature pyramid networks for object detection[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 2117-2125

[6] Dai J F, Qi H Z, Xiong Y W, *et al*. Deformable convolutional networks[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 764-773

[7] Redmon J, Divvala S, Girshick R, *et al*. You only look once: unified, real-time object detection[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 779-788

[8] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 7263-7271

[9] Liu W, Anguelov D, Erhan D, *et al*. SSD: single shot multibox detector[C] //Proceedings of the European Conference on Computer Vision. Heidelberg: Springer, 2016: 21-37

[10] Kong T, Sun F C, Yao A B, *et al*. RON: reverse connection with objectness prior networks for object detection[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 5244-5252

[11] Zhang S F, Wen L Y, Bian X, *et al*. Single-shot refinement neural network for object detection[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 4203-4212

[12] Chi C, Zhang S F, Xing J L, *et al*. Selective refinement network for high performance face detection[C] //Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2019: 8231-8238

[13] Lin T Y, Goyal P, Girshick R, *et al*. Focal loss for dense object detection[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 2999-3007

[14] Li B Y, Liu Y, Wang X G. Gradient harmonized single-stage detector[C] //Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2019: 8577-8584

[15] Cai Z W, Vasconcelos N. Cascade R-CNN: delving into high quality object detection[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 6154-6162

[16] Shrivastava A, Gupta A, Girshick R. Training region-based object detectors with online hard example mining[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 761-769

[17] He K M, Gkioxari G, Dollár P, *et al*. Mask R-CNN[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 2980-2988

[18] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[OL]. [2019-12-10] https://arxiv.org/abs/1409.1556

[19] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift[OL]. [2019-12-10]. https://arxiv.org/abs/1502.03167

[20] Chen Y T, Han C X, Wang N Y, *et al*. Revisiting feature alignment for one-stage object detection[OL]. [2019-12-10]. https://arxiv.org/abs/1908.01570

[21] He K M, Zhang X Y, Ren S Q, *et al*. Deep residual learning for image recognition[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 770-778

[22] Dai J F, Li Y, He K M, *et al*. R-FCN: object detection via region-based fully convolutional networks[C] //Proceedings of the Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2016: 379-387

[23] Zhu X Z, Hu H, Lin S, *et al*. Deformable ConvNets v2: more deformable, better results[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 9308-9316

[24] Fu C Y, Liu W, Ranga A, *et al*. DSSD: deconvolutional single shot detector[OL]. [2019-12-10]. https://arxiv.org/abs/1701.06659

[25] Liu S T, Huang D, Wang Y H. Receptive field block net for accurate and fast object detection[C] //Proceedings of the European Conference on Computer Vision. Heidelberg: Springer, 2018: 385-400

[26] Kim S W, Kook H K, Sun J Y, *et al*. Parallel feature pyramid network for object detection[C] //Proceedings of the European Conference on Computer Vision. Heidelberg: Springer, 2018: 234-250

[27] Law H, Deng J. CornerNet: detecting objects as paired keypoints[C] //Proceedings of the European Conference on Computer Vision. Heidelberg: Springer, 2018: 734-750

[28] Wang J Q, Chen K, Yang S, *et al*. Region proposal by guided anchoring[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 2960-2969

[29] Zhu C C, He Y H, Savvides M. Feature selective anchor-free module for single-shot object detection[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 840-849

[30] Yang Z, Liu S H, Hu H, *et al*. RepPoints: point set representation for object detection[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 9657-9666

[31] Tian Z, Shen C H, Chen H, *et al*. FCOS: fully convolutional one-stage object detection[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 9627-9636

[32] Cheng B W, Wei Y C, Shi H H, *et al*. Revisiting RCNN: on awakening the classification power of faster RCNN[C] //Proceedings of the European Conference on Computer Vision. Heidelberg: Springer, 2018: 473-490