计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(9):132−142 [doi: 10.15888/j.cnki.csa.009292]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 YOLOv5 改进的轻量化目标检测①

管嘉程[1,2], 任红卫[2], 周宋佳[2]

[1](吉林化工学院 信息与控制工程学院, 吉林 132022)
[2](广东石油化工学院 自动化学院, 茂名 525000)
通信作者: 任红卫, E-mail: renhongwei@gdupt.edu.cn

摘　要: 针对移动端目标检测算法需要模型参数量与计算量更少、推理速度更快和检测效果更好以及目标检测算法对于小目标误检、漏检及特征提取能力不足等问题, 提出一种基于 YOLOv5 改进的轻量化目标检测算法. 该算法使用轻量级网络 MobileNetV2 作为目标检测算法的骨干网络降低模型的参数量与计算量, 通过使用深度可分离卷积结合大卷积核的思想降低网络的计算量与参数量, 并提升了小目标的检测精度. 使用 GhostConv 来替换部分普通卷积, 进一步降低参数量与计算量. 本文算法在 VOC 竞赛数据集, COCO 竞赛数据集两份数据集上均进行了多次对比实验, 结果表明本文算法相比于其他模型参数量更小、计算量更小、推理速度更快以及检测精度更高.

关键词: 轻量化; 深度学习; 特征金字塔网络 (FPN); YOLOv5; 大核卷积

引用格式: 管嘉程,任红卫,周宋佳.基于 YOLOv5 改进的轻量化目标检测.计算机系统应用,2023,32(9):132–142. http://www.c-s-a.org.cn/1003-3254/9292.html

Improved Lightweight Target Detection Based on YOLOv5

GUAN Jia-Cheng[1,2], REN Hong-Wei[2], ZHOU Song-Jia[2]

[1](College of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin 132022, China)
[2](School of Automation, Guangdong University of Petrochemical Technology, Maoming 525000, China)

Abstract: Mobile target detection algorithms require fewer model parameters, less computation, faster reasoning speed, and better detection effects. The target detection algorithms are prone to false detection of small targets and missing detection and have insufficient ability for feature extraction. To this end, this study proposes a lightweight small target detection algorithm based on YOLOv5. In this algorithm, the lightweight network MobileNetV2 is used as the backbone network of the target detection algorithm to reduce the number of parameters and calculation amount of the model. The deep separable convolution combined with a large convolution kernel is applied to decline the number of parameters and calculation amount, and improve the detection accuracy of small targets. GhostConv is adopted to replace part of common convolution to further decrease the number of parameters and computation amount. Multiple comparison experiments are carried out on VOC competition data sets and COCO competition data sets. The results show that compared with other models, the proposed algorithm has fewer parameters, less computation, faster reasoning speed, and higher detection accuracy.

Key words: lightweight; deep learning; feature pyramid network (FPN); YOLOv5; large kernel convolution

① 基金项目: 广东省基础与应用基础研究基金 (2023A1515010168, 2019A1515010830); 广东省普通高校重点专项 (2022ZDZX1018); 茂名市科技计划 (2022S043); 广东石油化工学院博士启动项目 (2019BS001)
收稿时间: 2023-03-30; 修改时间: 2023-05-11; 采用时间: 2023-06-06; csa 在线出版时间: 2023-08-29
CNKI 网络首发时间: 2023-08-30

随着人工智能和深度学习思想的普及, 毫无疑问, 目标检测作为 CV 领域核心问题. 无论是在理论还是应用都进展飞速, 并广泛应用到了社会生活的各个方面, 诸如智能交通[1]、医疗辅助[2,3]、人像识别[4]、工业自动化[5]、运动识别[6] 等. 通过结合图像处理和深度学习等理论, 在图像中随机定位特定区域, 通过定位找出输入图像中目标物体的位置信息并确定目标框大小, 利用分类判断目标物体的类别. 且毫无疑问, 做好目标检测是实现目标跟踪, 场景理解, 事件检测等进阶视觉任务的首要任务.

现阶段目标检测的对象, 主要分为静态图像和动态视频. 视频目标检测以图像检测的理论为基石, 利用循环神经网络提取时序信息, 最后实现目标检测任务. 文献 [7] 总结罗列了近几年提出的诸多应用于静态图片的图像识别算法. 文献 [8] 总结罗列了这几年来的应用于视频文件的图像识别算法.

按目标检测算法的检测原理分类, 主要有两类: (1) 两阶段检测器, 诸如 SPPNET、R-CNN[9] 及其改进版[10–12]. (2) 一阶段检测器, 主要是 SSD[13]、YOLO 系列[14] 及其改进[15,16]. 两阶段精度稍高但速度略慢. 两阶段检测器的第 1 阶段先找出可能包含目标物体的建议框, 第 2 阶段对建议框进行分类, 进行预测; 一阶段检测器无须寻找建议框阶段, 直接确定物体的类别概率和位置坐标值, 经过单次检测即可直接得到最终的检测结果, 模型小、速度快、更具实用性.

近几年来, 因为 YOLOv5 既能满足实时性要求, 且能保持较高的检测精度, 使得其被广泛应用于各个领域. 且基于不同的操作环境和检测任务, 还可以选用不同模型大小的 YOLOv5. 其中 YOLOv5s 更因为其低计算量与高性能, 成为轻量化的理想候选者.

伴随着日益增长的移动端部署需求以及检测场景的多样性, 轻量化深度神经网络的要求迫在眉睫. 各种优秀的卷积网络的理论, 诸如 VGG[17]、ResNet[18]、DSC 卷积[19] 等, 都被融合到轻量化的网络中, 用来更有效的提取目标特征以及提升网络效率.

MobileNetV1[20] 结合深度可分离卷积的思想, 通过超参数使模型快速调节以适应特定的工作环境. 将其模型与时下主流的模型对比, 在模型大小和速度上, MobileNet 都展现出了极强优越性. MobileNetV2[21] 在 MobileNetV1 的基础上, 融合 ResNet 的思想, 提出倒残差的架构, 并使用一个线性的激活函数避免特征损失. 进一步降低模型大小, 且提升了准确率.

GhostNet[22] 发表于 CVPR2020 上的新颖的端侧神经网络, 通过组合少量卷积核与更廉价的线性变化操作代替常规卷积方式, 有效地改善了特征提取效率. 最近, 清华大学、旷视科技等机构的研究者在 CVPR2022 上提出了超大卷积核架构 RepLKNet[23], 利用少量大卷积核换取更大的感受野, 弥补了深层小卷积核模型有效感受野局限的缺陷.

文献 [24–27] 对 YOLOv5 模型做了改进, 一定程度上, 实现了轻量化, 却均没有用 COCO、PASCAL VOC 等一般数据集验证其性能. 文献 [28] 在 YOLOv5 上做了轻量化改进, 并用 PASCAL VOC 进行性能验证, 虽然降低了参数量, 计算量. 但是很大程度上, 牺牲了准确度和速度.

提高检测精度的代价是: 现代最先进的网络需要高计算资源, 超出了许多移动和嵌入式网络应用程序的能力. 为了解决 YOLOv5s 难以兼顾模型轻量化与模型检测精度以及对边界框的回归粗糙的问题. 本文对 YOLOv5s 进行改进, 提出 YOLO-MLK (you only look once-mobile large kernel) 目标检测算法用于移动端设备的目标检测任务, 主要贡献如下.

(1) 轻量级网络骨干. 使用参数量和计算量更小、移动端目标检测速度更快的 MobileNetV2 为基本架构, 替代原本 YOLOv5 的网络骨干, 降低网络的计算量, 提升模型的运算效率.

(2) 提出融合深度可分离大卷积的特征图金字塔模块 LKL-PAN. 通过拆分空间维度和通道维度的相关性, 减少卷积计算所需的参数个数, 提升卷积核参数的使用效率. 使用大卷积高效直观地增加感受野, 避免小卷积核的低效堆叠, 减少采样过程带来的特征损失, 进一步提升检测速度和检测精度.

(3) 优化激活函数. 引入 SIoU 激活函数, 增加检测框尺度的损失, 从而提升特征提取能力, 使得预测框更为精准, 进一步有提升网络的检测精度.

1 相关工作

1.1 YOLOv5

YOLOv5 是 YOLO 系列的经典算法. 按照其模型大小递增可分为 s、m、l、x 这 4 种, 所有模型均由输入端、Backbone、Neck、Head 构成. 在输入端部分, 首先对图片进行预处理, 在网络训练阶段使用 Mosaic

技术进行数据增强、自适应锚框计算以及自适应图片缩放: 在 Backbone 部分, YOLOv5 使用了改进的 CSP-Darknet 结构、Focus 下采样结构作为基准网络, 搭配 SPP 空间金字塔池化层更有效地提取特征信息; Neck 部分同样用到了 SPP 模块以及特征金字塔 FPN+PAN 模块, 实现细节与 Backbone 部分稍有不同, 进一步提升提取特征的多样性及鲁棒性. Head 用于完成目标检测结果的输出. 不同算法, Head 端的分支个数不同, 一般都有一个分类分支以及一个回归分支. YOLOv5 的基本架构如图 1 所示.

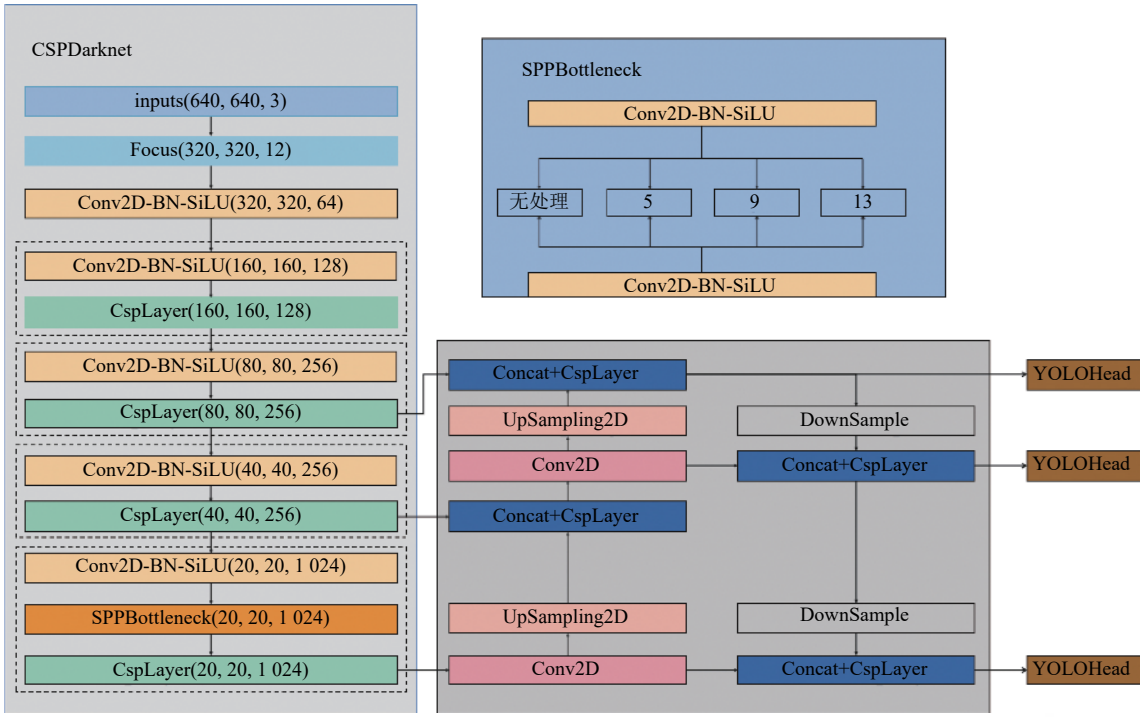


图 1 YOLOv5 算法架构

Conv2D 模块是复合卷积模块, 由卷积、BN 层和激活函数组成, 是 YOLOv5 的最基础模块. BN 层的目的是对数据做归一化处理, 防止训练网络的过程中出现梯度消失或爆炸.

Focus 模块, 首先将得到的图片进行切片操作. 将 RGB 三通道上的值每隔一个像素取下, 切分成 4 张特征图, 相当于将高、宽信息压缩到通道空间, 使得输入通道扩充为原先的 4 倍. 减少信息丢失的同时, 提升了网络的效率.

CspLayer 模块, 也常被称为 C3 模块. 特征图经过该模块会进入两个分支, 在一个分支中经过标准卷积层以及堆叠的 Bottleneck 模块; 另一分支中只经过一个标准卷积层, 最后将分别得到的特征图进行拼接. 该模块主要用于对残差特征进行学习.

SPP 模块为空间金字塔池化模块, 能够转换任意大小的特征图成为大小固定的特征向量. 当特征图经过 SPP 模块时, 首先经过卷积层减少通道数, 接着经过 3 个分支, 使用 3 个不同大小的卷积核进行池化下采样, 最后按通道数将池化结果与原本的特征图拼接. 通道数较原来稍有扩大, 但有效地提升了感受野.

1.2 深度可分离卷积

深度可分离卷积是改进标准卷积计算的算法, 其结构由逐通道卷积 (depthwise convolution) 和逐点卷积 (pointwise convolution) 组成.

逐通道卷积中, 每个通道的特征图都会通过一个卷积核进行卷积运算. 如图 2 前段部分所示, 此过程后, 得到的特征图的通道数与输入时的通道数一致.

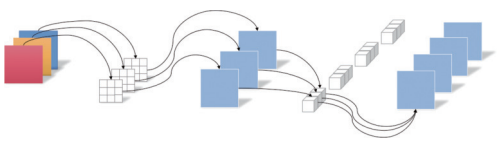


图 2 深度可分离卷积

逐点卷积与常规卷积的运算相似, 可以对特征图进行升维和降维操作, 其卷积核的尺寸为 1×1. 逐点卷积会将逐通道卷积取得的特征图在不同通道上进行加权组合, 生成最终的特征图.

深度可分离卷积通过转换空间维度和通道维度的信息, 提升卷积网络的效率, 降低卷积计算的参数量. 在检测任务中, 深度可分离卷积可以帮助模型有效降低计算量, 提高检测性能.

1.3 Ghost 卷积

Ghost 卷积的核心思想是将一般卷积拆分. 如图 3 所示, Ghost 卷积从少量非线性的卷积获取的特征上, 再使用线性卷积操作, 生成 Ghost 特征图. 接着将两段卷积得到特征图叠加, 得到更多通道数的特征图. 借此消除冗余特征, 轻量化模型计算.

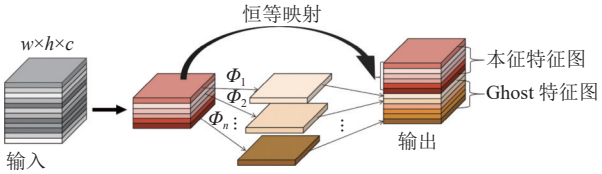


图 3 幽灵卷积

2 改进的 YOLOv5 算法

2.1 网络整体结构

本文对 YOLOv5 算法进行了改进, 改进后的 YOLO-MLK 模型网络结构如图 4 所示, 算法架构如表 1 所示. 首先, 使用轻量级网络 MobileNetV2 替代原本的骨干网络. 接着提出一种新的融合深度可分离大卷积的特征图金字塔网络 LKL-PAN. 最后, 替换了网络的损失函数, 使用 SIoU 作为网络的损失函数.

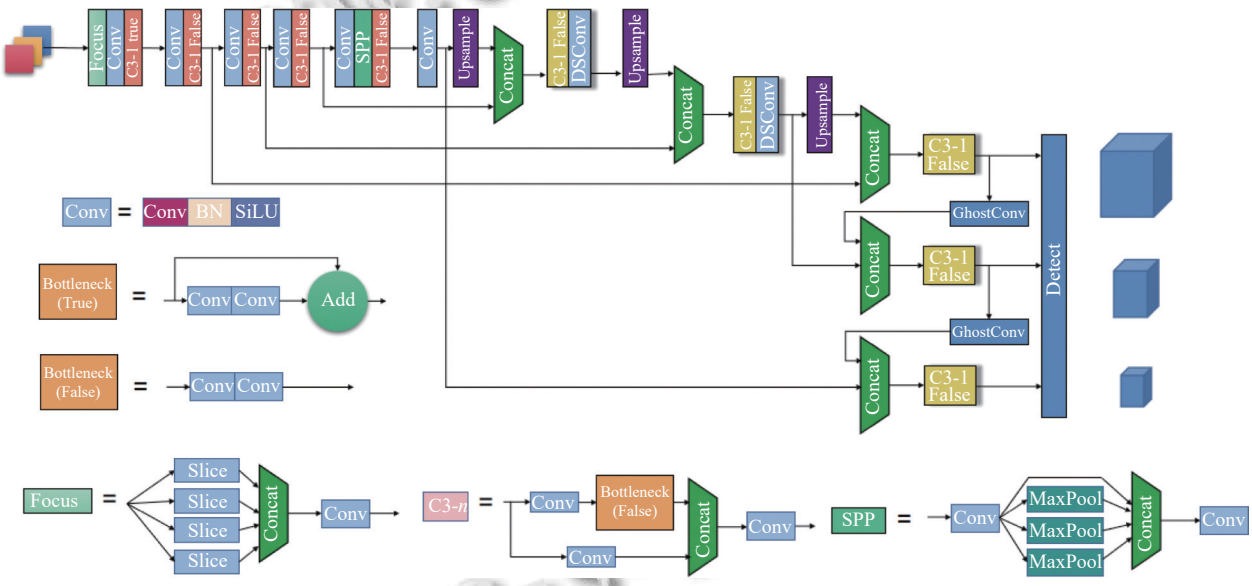


图 4 YOLO-MLK 模型架构

表 1 中, MobileNetV2-1 为 MobileNetV2 的 1–3 层, MobileNetV2-2 的 4 和 5 层, MobileNetV2-3 为 MobileNetV2 的 6–9 层, Upsample 为上采样, Concat 为数据拼接, SPPF 为快速空间金字塔池化模块, C3 为 YOLOv5 中的 C3 模块, DSConv 为深度可分离卷积, GhostConv 为幽灵卷积.

2.2 轻量化骨干网络

骨干网络是目标检测任务的基本特征提取器, 优质的骨干网络能够提取丰富的特征, 降低目标检测任务的复杂性, 提高目标检测网络的性能. YOLOv5 的骨干网络为 CSPDarknet53, 相比与 YOLOv4 的骨干网络具有参数量更小, 检测速度更快, 特征提取效率更高的优点. 但是 CSPDarknet53 并不能很好地适应移动端设备, 参数量、计算量和特征提取仍有提升的空间. 受目标检测网络模型轻量化的思想启发, 本文选取 MobileNetV2 替代 YOLOv5 中的 CSPDarknet 作为目标检测网络的骨干网络, 降低模型的计算量和计算量, 提高模型的特征提取效率, 模块组成如表 2 所示. 其中, Input 表示输入的特征图大小, Operator 表示执行相应的操作, t 表示瓶颈层内部升维的倍数, c 表示输出特征的维数, n 表示该瓶颈层重复的次数, s 表示瓶颈层第 1 个卷积操作的步幅, Conv2D 表示标准卷积

模块, Bottle-neck 表示瓶颈层模块, avgpool 表示平均池化操作.

表 1 YOLO-MLK 算法的架构

层数	From	参数量	模块名称
1	−1	55 488	MobileNetV2-1
2	−1	487 040	MobileNetV2-2
3	−1	1 681 344	MobileNetV2-3
4	−1	2 132 224	SPPF
5	−1	87 040	DSConv
6	−1	0	Upsample
7	[−1, 2]	0	Concat
8	−1	321 024	C3
9	−1	60 160	DSConv
10	−1	0	Upsample
11	[−1, 1]	0	Concat
12	−1	78 592	C3
13	−1	75 584	GhostConv
14	[−1, 9]	0	Concat
15	−1	329 216	C3
16	−1	298 624	GhostConv
17	[−1, 4]	0	Concat
18	−1	118 720	C3
19	−1	67 425	Detect

表 2 MobileNetV2 算法的架构

Input	Operator	t	c	n	s
$224^2\times3$	Conv2D	—	32	1	2
$112^2\times32$	Bottleneck	1	16	1	1
$112^2\times16$	Bottleneck	6	24	2	2
$56^2\times24$	Bottleneck	6	32	3	2
$28^2\times32$	Bottleneck	6	64	4	2
$14^2\times64$	Bottleneck	6	96	3	1
$14^2\times96$	Bottleneck	6	160	3	2
$7^2\times160$	Bottleneck	6	320	1	1
$7^2\times320$	Conv2D 1×1	—	1 280	1	1
$7^2\times1280$	avgpool 7×7	—	—	1	—
1×1×1280	Conv2D 1×1	—	k	—	

相比于 MobileNetV1 网络, MobileNetV2 网络优化了瓶颈层 (Bottleneck) 的结构, 提高网络的特征提取效率和能力. MobileNetV2 的瓶颈层使用了倒残差结构和线性瓶颈层的思想, 其由扩展层、逐通道层和投影层构成, 如图 5 所示.

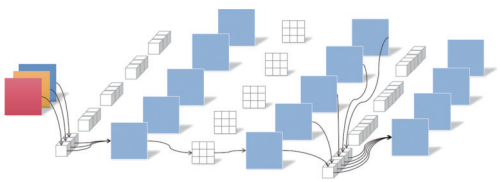


图 5 MobileNetV2 的 Bottleneck

每个瓶颈层首先在扩展层使用 1×1 卷积将低维特征映射到高维空间, 并通过 ReLU6 激活函数激活. 然后经过逐通道层时, 使用 3×3 的逐通道卷积将高维特征映射到高维空间, 提取到足够多的整体信息, 并通过 ReLU6 函数激活. 最后经过投影层时, 使用 1×1 的普通卷积, 将高维特征映射回低维空间去, 并使用线性函数激活, 可以有效防止高维特征映射到低维空间时丢失提取出的特征. 这样的瓶颈层结构可以保证在进行残差连接的时候, 相互连接的都是低维度的, 来减少计算量.

2.3 融合深度可分离大卷积的特征图金字塔模块

FPN 能够融合不同层的特征信息, 基本不增加模型计算量, 且能有效提升目标检测的性能. 尽管 FPN 只提出了短短几年, 却已被广泛用于机器视觉领域, 用于不同尺度的特征融合. 越来越多的多尺度特征融合网络被提出, 诸如 NAS-FPN、M2det 以及 PANet 等, 都展现出了优异的效果.

自底向上的前向传播结构和自顶而下的上采样结构, 通过横向连接进行路径整合, 使得 YOLOv5 中的 FPN+PAN 具备了更优的多尺度融合结果. 但其劣势也十分明显, 模型结构更加冗余复杂, 从而使得特征主干网络的原始输出特征无法被有效地提取.

深度可分离大卷积通过直观地增加目标检测任务的感受野, 避免了小卷积核不断堆叠的低效陷阱, 并且深度可分离通过对普通卷积进行纵向和横向的解耦, 降低参数量与计算量, 提升了卷积的效率. Ghost 卷积首先利用少量的卷积核对输入特征图进行特征提取, 然后进一步地对这部分特征图进行更价廉的线性变化运算, 最后通过 Concatenation (拼接操作) 生成最终的特征图. 这个方法减少了非关键特征的学习成本: 即通过组合少量卷积核与更廉价的线性变化操作代替常规卷积方式, 从而有效降低对计算资源需求的同时, 并不影响模型的性能.

为此, 提出了一种简单但高效的融合深度可分离大卷积与 Ghost 卷积的特征图金字塔网络, 称为 LKL-PAN (large kernel lightweight aggregation network).

LKL-PAN 使用深度可分离卷积结合大卷积并结合 Ghost 卷积对颈部网络进行了改进, 在 PAN 部分使用深度可分离卷积结合大核卷积的思想直接提升了目标的感受野, 并且对算法的参数量与计算量进行控制, 其结构如图 6 所示.

提出的特征图金字塔网络具体实施过程如下.

(1) 使用卷积核大小为 13×13 深度可分离大卷积

替换原本的 FPN 结构中下采样阶段卷积, 以此扩大特征提取有效感受野, 避免了小卷积核不断堆叠的低效陷阱, 提升网络的特征提取能力.

(2) 使用 3×3 Ghost 卷积替换原本 PAN 结构上采样阶段卷积, 从而实现轻量化特征提取网络. Ghost 卷积使用少量的卷积核对输入特征图进行特征提取, 通过更价廉的线性变化运算并使用拼接操作生成最终的特征图, 减少了非关键特征的学习成本.

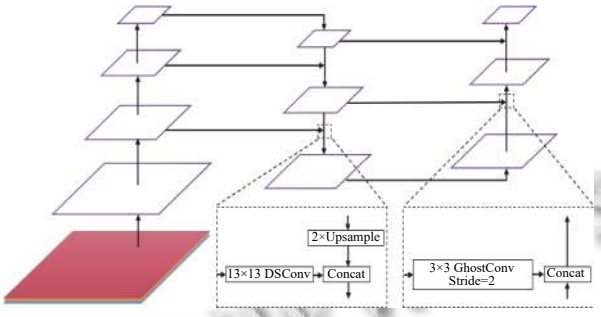


图 6 LKL-PAN 结构

如何合理选择深度可分离大卷积核的尺度, 从而提取更有效的特征? 在综合考虑模型参数量与计算量后, 本文选择使用卷积核大小为 21、13 和 7 的深度可分离卷积替代 Neck 中下采样的阶段的普通卷积, 并在 VOC 数据集上使用 *mAP*@0.5 这个指标来衡量模型的效果.

卷积核组合表示两个下采样阶段的深度可分离卷积的卷积核大小, 参数量表示模型总体参数量大小, 计算量表示模型总体计算量大小, *mAP*@0.5 表示预测精度.

从表 3 可知, 使用 21+13 的卷积核组合效果最好, 但模型参数量较大. 使用 13+13 的卷积核组合效果与 21+21 的卷积核组合精度排并列第二. 综合考虑模型的参数量、计算量与预测精度后, 本文最终选择 13+13 的卷积核组合.

表 3 不同尺度深度可分离大卷积核实验

卷积核组合	参数量	计算量 (GFLOPs)	*mAP*@0.5 (%)
21+21	7.2	12.4	83.2
21+13	7.1	12.2	83.4
13+21	7.0	12.3	83.1
13+13	6.9	12.1	83.2
13+7	6.9	12.0	83.0
7+13	6.8	12.1	82.8
7+7	6.8	12.0	82.8

2.4 损失函数

YOLOv5 中的损失函数一共由 3 部分组成, 分别是分类损失、定位损失、置信度损失. 分类损失用于评估预测框及对应分类的正确程度; 定位损失用来表示预测框与真实目标框两者间的误差大小; 置信度损失表示锚框中目标物体是否存在的条件损失.

CIoU 损失函数考虑了检测框尺度和检测框长和宽的 loss, 这使预测框更加的符合真实框. 但未解决检测框纵横比描述使用的是相对值, 且 CIoU 损失函数中并没有考虑检测框的角度问题, 这也会影响模型在训练过程中的回归.

为减小上述 CIoU 损失函数在实际应用中暴露的问题, 本研究采用了 SIoU 损失函数.

SIoU 在 CIoU 的基础上通过计算检测框宽高的差异值取代了纵横比, 解决了检测框纵横比描述使用的是相对值的问题, 优化预测框的大小的确定. 并考虑了检测框角度损失对于确定最终预测框的影响, 使得预测框位置更加准确, 也优化模型在训练过程中的回归. SIoU 损失函数包含 4 个部分: 形状损失、*IoU*损失、距离损失以及角度损失. 计算公式如下:

$$L_{\mathrm{SIoU}} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{1}$$

其中, Ω是形状损失, Δ是距离损失, *IoU*为目标准确度损失. SIoU 对距离损失做了重新定义, 把角度损失Λ也纳入了考虑范畴. 如式 (2)–式 (4) 所示:

$$\Delta = \sum_{t=x,y} (1 - \mathrm{e}^{-\gamma\rho_t}) \tag{2}$$

$$\Lambda = 1 - 2 \times \sin^2\left(\arcsin(x) - \frac{\pi}{4}\right) \tag{3}$$

$$x = \frac{C_h}{\sigma} = \sin(\alpha) \tag{4}$$

其中, σ代表目标框 B 与目标框 B^{GT} 的中心连线, C_h代表垂直距离等于σ的长度取正弦值, 如图 7 所示.

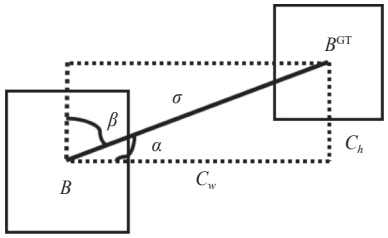


图 7 SIoU 角度损失示意图

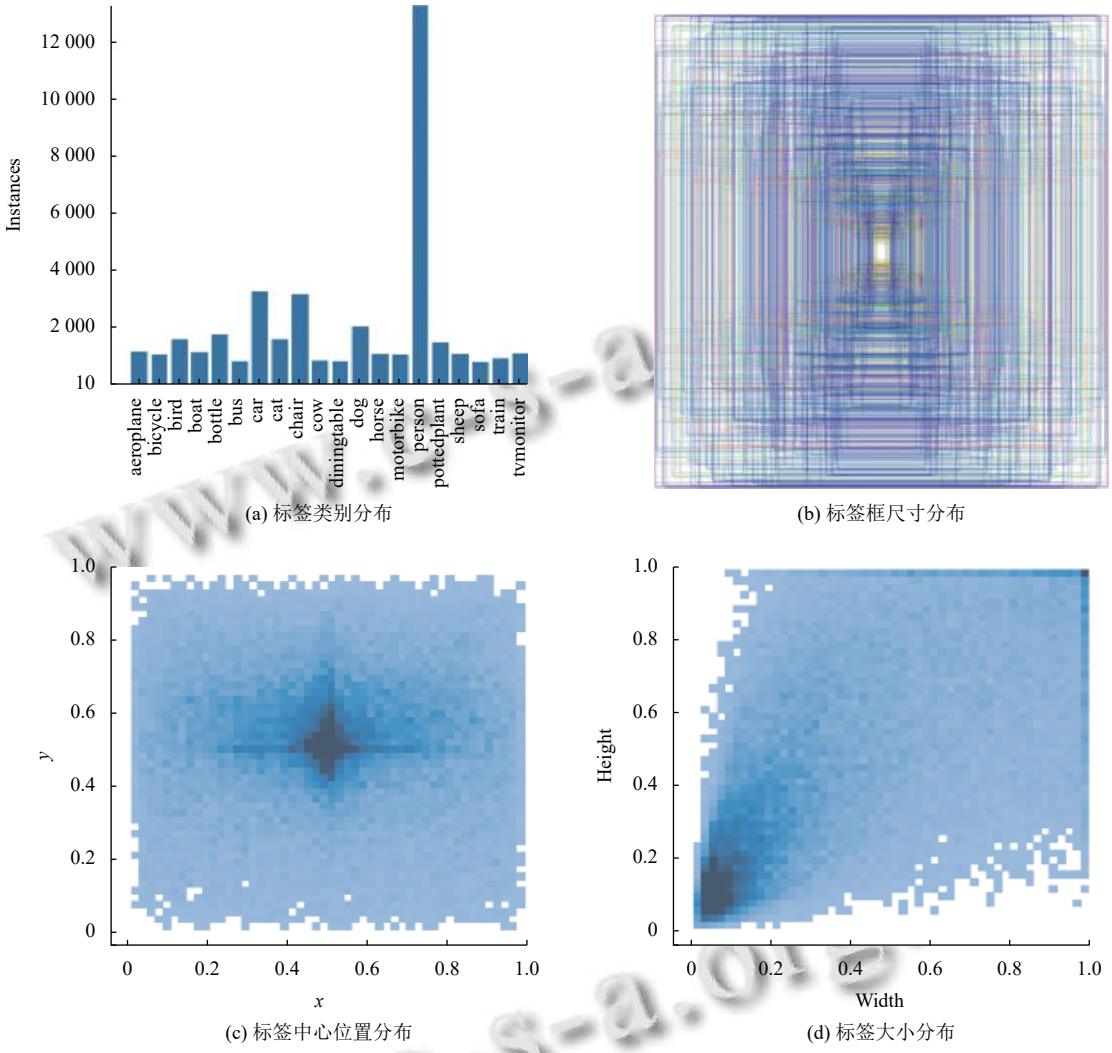
3 实验与结果分析

3.1 数据集

3.1.1 PASCAL VOC 数据集

PASCAL VOC (the PASCAL visual object classes) 是世界闻名的 CV 挑战赛. 本研究选取了 PASCAL

VOC 2007 和 2012 (即 VOC 2007+VOC 2012) 数据集进行实验, 整个数据集总共包含 4 大类和 20 小类. 训练集部分, 选用了 VOC 2007 以及 VOC 2012 数据集的 train 和 val 部分, 总共包含图片 16 551 张; 测试集部分, 选用 VOC 2007 数据集的 test 部分, 总共包含图片 4 952 张. 图 8 是数据可视化分析.



(a) 标签类别分布 (b) 标签框尺寸分布

(c) 标签中心位置分布 (d) 标签大小分布

图 8 VOC 数据集可视化分析

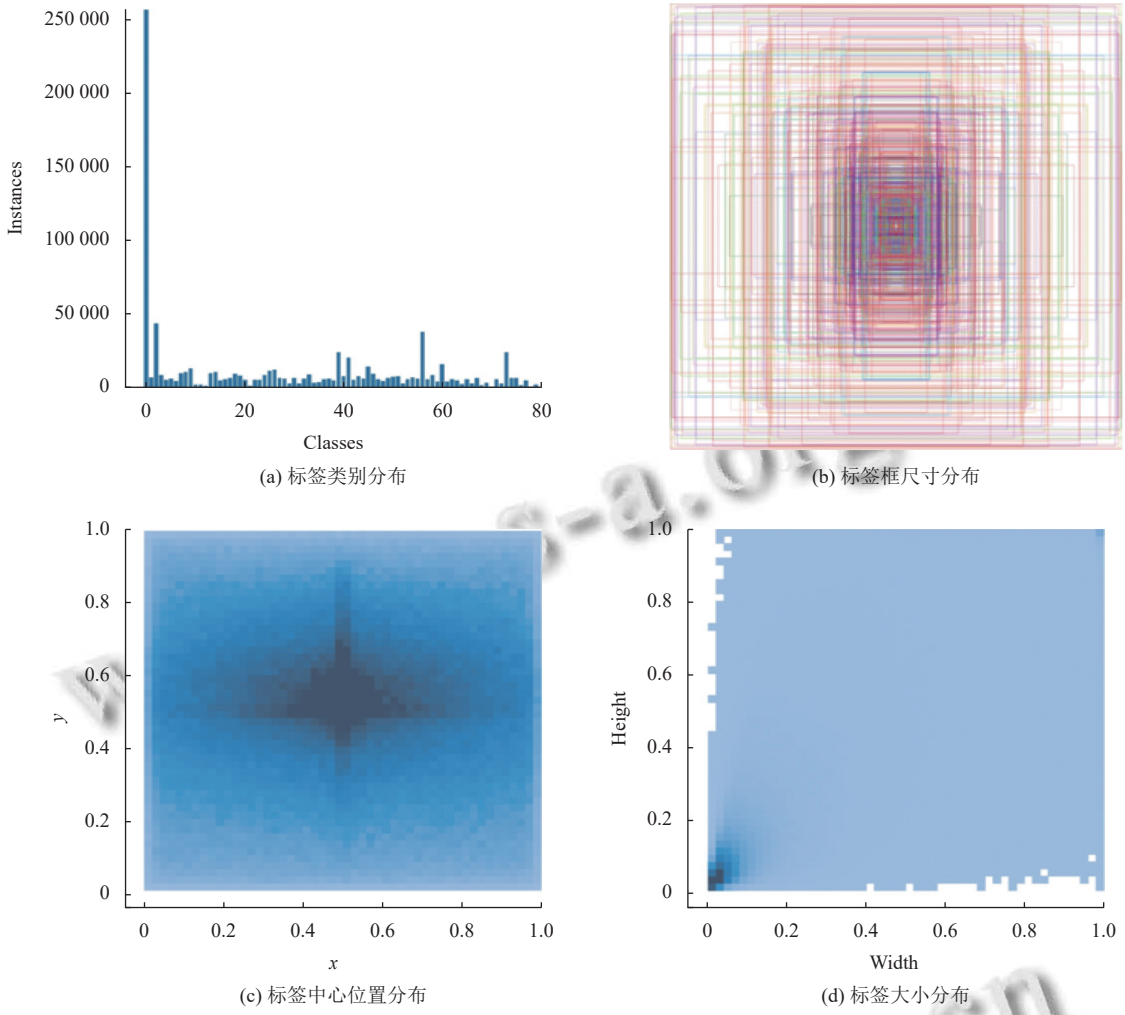
3.1.2 COCO 数据集

MS COCO (Microsoft common objects in context) 是机器视觉领域最权威和关注度最高的的比赛之一. 该数据集主要从复杂的日常场景中截取, 是同时可用于语义分割, 图像标题生成和图像检测的大型数据集. 作为目前有语义分割的数据集, 其中收录了超过 330k 张图像 (其中超 200k 张已标注过), 目标数超 150 万个, 80 个目标类别 (object categories: 火车、船、猫等), 91 种无明确边界的材料类别 (stuff categories: 街道、墙、天空等) 以及带关键点标注的 25 万个行人影像. 以下是 COCO 数据集的数据可视化分析, 如图 9 所示.

3.1.3 网络设置与训练

实验所使用硬件配置如表 4 所示.

在网络训练前, 把数据集设置为训练集、验证集和测试集. 实验总迭代设置为 300 次, 前 3 次迭代用作预训练, 学习率调整采用梯度下降 (SGD) 策略. 预热结束, 采用余弦退火策略. 网络训练中, batch_size 的参数调整为 8. 训练完 YOLOv5 模型后, 接着训练 YOLO-MLK 模型, 将 YOLOv5 的部分权重转移到 YOLO-MLK 上, 可节省大量的训练时间. 同上, 实验总迭代 300 次, 其他参数保持一致.



(a) 标签类别分布 (b) 标签框尺寸分布

(c) 标签中心位置分布 (d) 标签大小分布

图 9 COCO 数据集可视化分析

表 4 实验硬件配置

参数	实验环境
操作系统	Ubuntu 18.04
CPU	11th Gen Intel® Core™ i7-11700F CPU @ 2.50 GHz
GPU	NVIDIA T4×3
内存	64 GB
Python	3.8
深度学习框架	PyTorch 1.11.1, CUDA 11.1

3.2 评价指标与实验结构分析

3.2.1 评价指标

为验证算法的性能, 本研究选用了几项目标检测中常用的评价标准, 参数量, 模型大小, 计算量, *mAP*@0.5 来衡量本文提出的 YOLO-MLK 算法模型.

模型大小, 参数量和计算量用作衡量模型的复杂程度和网络的深度, *mAP*@0.5 主要体现神经网络的检测能力是否准确. *mAP*@0.5:0.95 要求更高的 *IoU* 阈值, 在准确召回的基础上, 用于衡量目标定位效果和边框回归的是否精准. 平均精度均值 *mAP* 中的 *P*-*R* 曲线, 即平均精度 *AP*. 计算公式如下.

(1) 精确率 (*P*) 和召回率 (*R*):

$$P = \frac{TP}{TP+FP} \cdot 100\% \tag{5}$$

$$R = \frac{TP}{TP+FN} \cdot 100\% \tag{6}$$

其中, *TP* (true positives) 代表被准确识别出的目标数, *FP* (false positives) 代表错检的目标数, *FN* (false negatives) 表示未被检出的目标数.

(2) 平均精度和平均精度均值:

$$AP = \int_0^1 P(R)dR \tag{7}$$

$$mAP = \frac{\sum P_A}{Nc} \tag{8}$$

其中, Nc 表示检测目标类别的数量, P_A 表示单个类别计算出的平均精度. 模型的 P-R 曲线可由得到的实验数据绘制, 曲线的面积即为 AP 值. mAP 表示全部目标类别的 AP 取平均值的结果. mAP 值越高, 越趋近于 1. 表示神经网络的识别能力越强.

3.2.2 VOC 数据集实验分析

为了验证本文所提算法 YOLO-MLK 的网络性能, 选取了 YOLOv3、YOLOv5s、YOLOv5-MobileNetV3-Large、YOLOv4-MobileNetV2、YOLOv4-MobileNetV3-Large、YOLOv4-EEEA-Net-C2、SSD、Faster、Cascade 等模型在 VOC 数据集上进行对比. 实验结果如表 5 所示. 可见, YOLO-MLK 不仅模型的复杂度更小, 在精度上亦优势明显. 其 mAP@0.5 是对比的众多模型中最优的.

表 5 VOC 数据集实验分析

模型	Params (M)	模型大小 (M)	计算量 (GFLOPs)	mAP@0.5 (%)
YOLOv5s (2020)	7.1	14.2	16.1	82.0
YOLOv5-MobileNetV3-Large (2022)	7.45	14.9	11.7	81.6
YOLOv4-MobileNetV2 (2021)	46.3	92.6	8.7	81.5
YOLOv4-MobileNetV3-Large (2021)	47.3	94.6	8.5	78.9
YOLOv4-EEEA-Net-C2 (2021)	31.5	63.0	5.5	81.8
SSD	41.1	82.2	387.9	76.7
YOLOv3	62.0	134.0	156.4	82.4
Ours	6.9	13.8	12.1	**83.2**

与参数量更大, 计算量相近 YOLOv5-MobileNetV3-Large 相比, YOLO-MLK 在 mAP@0.5 上有 1.4 个百分点的提升. 与基于 YOLOv4 框架的 MobileNetV2、MobileNetV3-Large、EEEA-Net-C2 轻量级主干目标检测算法对比, YOLO-MLK 无论在参数量还是检测精度方面都具明显优势. 与 YOLOv5s 对比, 参数量 mAP@0.5 上提高 1.1%. 在计算量这个指标上, YOLO-MLK 也是优于众多目标检测算法. YOLO-MLK 优化了模型的时间、空间复杂度, 且明显提升了检测精度. 并使其对硬件需求更低, 更适用于成本相对较低的工业检测问题.

3.2.3 COCO 数据集实验分析

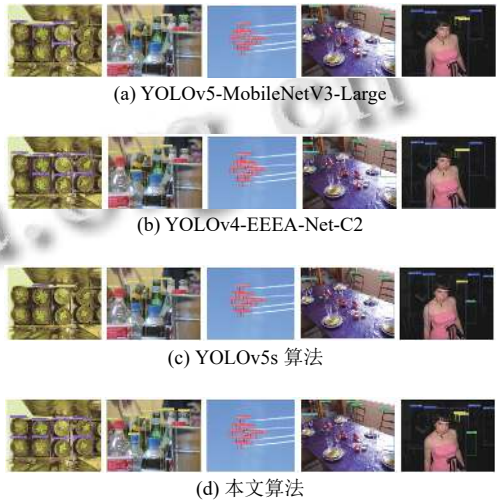
为了进一步验证 YOLO-MLK 的网络性能, 将其与 YOLOv5s、SSD、YOLOX-Tiny、YOLOv6-N、YOLOv7-Tiny 等进行比较, 实验结果如表 6 所示. 显而易见, YOLO-MLK 在 mAP@0.5:0.95 值上, 优于其他目标检测算法. 在与原始 YOLOv5 对比中, YOLO-MLK 算法在模型参数量和计算量显著降低的情况下, mAP@0.5:0.95 提升了 0.3%. 在与最新的 YOLOv7-Tiny 算法的对比中, YOLO-MLK 在模型计算量明显降低更有优势, 并且 mAP@0.5:0.95 略高. 这证明了 YOLO-MLK 具备更优越的检测能力.

表 6 COCO 数据集实验分析

模型	Params (M)	模型大小 (M)	计算量 (GFLOPs)	mAP@0.5:0.95 (%)
YOLOv5s (2020)	7.2	14.4	16.5	37.2
SSD	36.1	62.2	—	25.1
YOLOv4-Tiny (2020)	6.1	12.2	—	21.7
YOLOX-Tiny (2021)	6.5	13.0	5.1	32.8
YOLOv6-N (2022)	4.3	8.6	11.7	35.9
YOLOv7-Tiny (2022)	6.2	12.4	13.7	37.4
Ours	7.1	14.2	12.8	**37.5**

3.3 定性评价

本研究选取了 5 组 VOC 测试集中的目标图片对 YOLOv5-MobileNetV3-Large、YOLOv5s、YOLOv4-EEEA-Net-C2 和 YOLO-MLK 的检测效果进行定性评价, 对比结果如图 10 所示.



(a) YOLOv5-MobileNetV3-Large

(b) YOLOv4-EEEA-Net-C2

(c) YOLOv5s 算法

(d) 本文算法

图 10 其他算法与本文算法效果对比图

在第 1 组先验目标较少的实验图片中, 其他的模型均出现一定程度的遗漏目标的情况, 而 YOLO-MLK 准确无误地识别出了所有目标. 在第 2 组目标较为多, 且有明显遮挡情况的实验图片中, YOLOv5-MobileNetV3-Large 中出现了大量很大程度的漏检, YOLOv4-EEEA-Net-C2 在大量漏检的情况下, 还出现了错检. YOLOv5s

中的漏检相对改善. YOLO-MLK检测出了更多的正确目标, 表现出了更优越检测性能. 在第 3 组目标密集但并无遮挡的实验图片中, 其他的算法模型均出现了不同程度的错检, 而 YOLO-MLK 准确鉴别出所有目标, 且无错检与漏检的情况出现. 在第 4 组实验图片的目标数较少, 但有部分位于图像边缘且类别特征并不完整的目标. YOLOv5-MobileNetV3-Large、YOLOv4-EEEA-Net-C2、YOLOv5s 均有边缘目标未被识别的情况发生, YOLO-MLK 明显改善了边缘目标的漏检情况. 第 5 组实验图片中的目标较多, 大中小目标均存在, YOLOv5-MobileNetV3-Large、YOLOv5s、YOLOv4-EEEA-Net-C2 在小目标和中等目标的检测中, 都出现了漏检, 而 YOLO-MLK 明显改善了这种情况. 总的来说, YOLO-MLK 在与其他一些主流轻量化模型对比中, 对小目标的检测能力更出众, 漏检、错检率减低, 检测精度明显提升. 证明 YOLO-MLK 更有效的提取了语义信息, 从而表现出来更优秀的检测能力.

3.4 消融实验

为了探究单个改进模块对整体算法的提升效果, 本节基于 VOC 数据集进行了消融实验. 首先用原始 YOLOv5s 算法为基础, 接着依次进行了 7 项消融实验, 表 7 列出了消融实验的结果. 其中, MobileNetV2 代表 MobileNetV2 轻量级骨干网络, LKL-PAN 代表本文提出的特征提取金字塔结构, SIoU 代表本文所使用的改进损失函数.

如表 7 所示, 用 MobileNetV2 替代原 YOLOv5 网络主干, 模型参数量增加了 1.41%, 计算量减少了 22.98%, *mAP*@0.5 值上升 0.2%. 证明 MobileNetV2 主干网络, 大幅减低计算量的同时, 在特征提取能力方面也有提升; 在引入 LKL-PAN 模块的情况下, 模型的参数量减少了 9.86%, 计算量下降 3.73%, *mAP*@0.5 提升 0.7%. 显然 LKL-PAN 相较原本的 FPN+PAN 模块, 在参数量下降的同时, 其颈部网络的特征提取与融合亦有明显提升, 从而有效提升了模型的检测效率; 在引入 SIoU 损失函数的情况下, *mAP*@0.5 提升了 0.3%. 证明 SIoU 很大程度上, 缓解了真实框与预测框之间不匹配的问题. 最终 YOLO-MLK 相较 YOLOv5s, 模型参数量减少了 2.82%, 计算量下降 24.84%, *mAP*@0.5 提升 0.7%. YOLO-MLK 算法加强了对目标框的拟合能力, 且对硬件的要求更低. 更适用于对 *IoU* 值要求更高, 检测精度更高的检测任务中.

表 7 消融实验

组别	MobileNetV2	LKL-PAN	SIoU	Params (M)	计算量 (GFLOPs)	*mAP*@0.5 (%)
1	×	×	×	7.1	16.1	82.0
2	√	×	×	7.2	12.4	82.5
3	×	√	×	6.4	15.5	83.2
4	×	×	√	7.1	16.1	82.2
5	√	√	×	6.9	12.1	83.1
6	√	×	√	7.2	12.4	82.8
7	×	√	√	6.8	15.8	83.3
8	√	√	√	6.9	12.1	83.2

4 实验与结果分析

针对移动端检测任务需求的日益增长, 以及其有限的硬件配置问题, 本研究提出了一种改进的轻量化模型 YOLO-MLK. 采用 MobileNetV2 网络作为骨干网络, 实现模型计算量的轻量化; 采用 LKL-PAN 结构, 对颈部网络进行轻量化, 并获取更大且有效的感受野, 提升特征提取能力; 引入 SIoU 损失函数, 缓解真实框与预测框之间不匹配的问题, 提升目标框拟合能力, 减小检测框损失, 获取更为准确的预测框.

本研究在 VOC 数据集和 COCO 数据集上做了多组对比实验, 减少模型了参数量、计算量, *mAP* 也得到了明显提升, 证明了 YOLO-MLK 的有效性. 为了轻量化神经网络算法, 降低模型复杂度, 从而满足移动端需求. 后续工作将对网络进行剪枝、蒸馏操作; 增加检测头, 提升对小目标的特征提取能力; 改进头部网络的上采样结构, 更高效地提取特征, 提升检测精度.

参考文献

1 Zhang LL, Lin L, Liang XD, *et al*. Is Faster R-CNN doing well for pedestrian detection? Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 443–457.

2 Liu Y, Ma Z, Liu XM, *et al*. Privacy-preserving object detection for medical images with Faster R-CNN. IEEE Transactions on Information Forensics and Security, 2019, 17: 69–84.

3 Jaeger PF, Kohl SAA, Bickelhaupt S, *et al*. Retina U-Net: Embarrassingly simple exploitation of segmentation supervision for medical object detection. Proceedings of the 2020 Machine Learning for Health Workshop. Vancouver: PMLR, 2020. 171–183.

4 Najibi M, Samangouei P, Chellappa R, *et al*. SSH: Single stage headless face detector. Proceedings of the 2017 IEEE

International Conference on Computer Vision. Venice: IEEE, 2017. 4885–4894.
5 李成严, 马金涛, 赵帅. 基于空间域注意力机制的车间人员检测方法. 哈尔滨理工大学学报, 2022, 27(2): 92–98.
6 Raghunandan A, Mohana, Raghav P, *et al*. Object detection algorithms for video surveillance applications. Proceedings of the 2018 International Conference on Communication and Signal Processing (ICCSP). Chennai: IEEE, 2018. 563–568.
7 罗会兰, 陈鸿坤. 基于深度学习的目标检测研究综述. 电子学报, 2020, 48(6): 1230–1239. [doi: 10.3969/j.issn.0372-2112.2020.06.026]
8 王迪聪, 白晨帅, 邬开俊. 基于深度学习的视频目标检测综述. 计算机科学与探索, 2021, 15(9): 1563–1577. [doi: 10.3778/j.issn.1673-9418.2103107]
9 Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus: IEEE, 2014. 580–587.
10 Girshick R. Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). Santiago: IEEE, 2015. 1440–1448.
11 Ren SQ, He KM, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2015. 91–99.
12 Lin TY, Dollár P, Girshick R, *et al*. Feature pyramid networks for object detection. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 936–944.
13 Lin W, Anguelov D, Erhan D, *et al*. SSD: Single shot multibox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 21–37.
14 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 779–788.
15 Redmon J, Farhad A. YOLO9000: Better, faster, stronger. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 6517–6525.
16 Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv:1804.02767, 2018.
17 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Proceedings of the 3rd International Conference on Learning Representations. San Diego: ICLR, 2014.
18 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016. 770–778.
19 Sifre L, Mallat S. Rigid-motion scattering for texture classification. arXiv:1403.1687, 2014.
20 Howard AG, Zhu ML, Chen B, *et al*. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861, 2017.
21 Sandler M, Howard AG, Zhu ML, *et al*. MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 4510–4520.
22 Han K, Wang YH, Tian Q, *et al*. GhostNet: More features from cheap operations. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle: IEEE, 2020. 1577–1586.
23 Ding XH, Zhang XY, Han JG, *et al*. Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs. Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022. 11953–11965.
24 彭成, 张乔虹, 唐朝晖, 等. 基于 YOLOv5 增强模型的口罩佩戴检测方法研究. 计算机工程, 2022, 48(4): 39–49.
25 杨小冈, 高凡, 卢瑞涛, 等. 基于改进 YOLOv5 的轻量化航空目标检测方法. 信息与控制, 2022, 51(3): 361–368. [doi: 10.13976/j.cnki.xk.2021.1240]
26 林森, 刘美怡, 陶志勇. 采用注意力机制与改进 YOLOv5 的水下珍品检测. 农业工程学报, 2021, 37(18): 307–314. [doi: 10.11975/j.issn.1002-6819.2021.18.035]
27 钱坤, 李晨瑄, 陈美杉, 等. 基于 YOLOv5 的舰船目标及关键部位检测算法. 系统工程与电子技术, 2022, 44(6): 1823–1832. [doi: 10.12305/j.issn.1001-506X.2022.06.07]
28 邱天衡, 王玲, 王鹏, 等. 基于改进 YOLOv5 的目标检测算法研究. 计算机工程与应用, 2022, 58(13): 63–73. [doi: 10.3778/j.issn.1002-8331.2202-0093]

(校对责编: 孙君艳)