

DID-YOLO:一种适用于嵌入式设备的 移动机器人目标检测算法

章佳琪¹, 肖建^{2*}

(1. 南京邮电大学 电子与光学工程学院、柔性电子(未来技术)学院, 江苏 南京 210046;
2. 南京邮电大学 集成电路科学与工程学院, 江苏 南京 210046)

摘要:近些年来,目标检测算法在移动机器人环境感知领域表现出了突出的性能。但是目标检测算法存在模型庞大和计算复杂的问题,制约了目标检测算法在移动嵌入式设备上的部署和发展。YOLO是一种单阶段的目标检测算法,具有较高的准确度和较快的运行速度。该文提出了一种基于YOLOv5s改进后适用于嵌入式设备的移动机器人目标检测算法DID-YOLO。首先,使用深度可分离卷积和倒置残差模块对YOLOv5s的backbone网络进行重构,降低模型复杂度和计算量,达到轻量化的目的;其次,利用特征层和输出层结合的知识蒸馏训练提高重构后目标检测网络的精度。在目标检测通用数据集PASCAL VOC上实验表明:DID-YOLO模型尺寸为3.63 MB,相较原网络模型尺寸减小了48.65%;经过特征层和输出层蒸馏后,DID-YOLO的mAP@0.5提升至73.83%;DID-YOLO在Jetson AGX Xavier上实现了每秒31.2帧的实时图像处理速度。提出的DID-YOLO性能显著,满足了移动机器人嵌入式平台的实时高精度检测需求。

关键词:移动机器人;目标检测;嵌入式设备;轻量化;知识蒸馏

中图分类号:TP249

文献标识码:A

文章编号:1673-629X(2023)10-0008-07

doi:10.3969/j.issn.1673-629X.2023.10.002

DID-YOLO: A Mobile Robot Target Detection Algorithm for Embedded Devices

ZHANG Jia-qi¹, XIAO Jian^{2*}

(1. School of Electronic and Optical Engineering & School of Flexible Electronics (Future Technology),
Nanjing University of Posts and Telecommunications, Nanjing 210046, China;
2. School of Integrated Circuit Science and Engineering, Nanjing University of Posts and
Telecommunications, Nanjing 210046, China)

Abstract: In recent years, object detection algorithm has shown outstanding performance in the field of mobile robot environment perception. However, the problem of huge model and complex calculation of target detection algorithm restricts the deployment and development of such algorithm on mobile embedded devices. YOLO is a single-stage target detection algorithm with high accuracy and fast running speed. We propose an improved DID-YOLO mobile robot target detection algorithm based on YOLOv5s, which is suitable for embedded devices. Firstly, deep separable convolution and inverted residual modules are used to reconstruct the backbone network of YOLOv5s to reduce model complexity and computational load for lightweight. Secondly, knowledge distillation training which combines the feature layer and the output layer is used to improve the accuracy of the reconstructed target detection network. Experiments on PASCAL VOC show that the size of DID-YOLO model is 3.63 MB, which is 48.65% smaller than that of the original network model. After distillation of the feature layer and the output layer, mAP@0.5 of DID-YOLO increased to 73.83%. DID-YOLO achieved a real-time image processing speed of 31.2 frames per second on Jetson AGX Xavier. The DID-YOLO proposed has remarkable performance and can meet the requirements of real-time and high-precision detection of embedded mobile robot platforms.

Key words: mobile robot; object detection; embedded device; lightweight; knowledge distillation

收稿日期:2023-01-11

修回日期:2023-05-11

基金项目:国家自然科学基金项目(61974073)

作者简介:章佳琪(1998-),男,硕士研究生,研究方向为嵌入式系统应用;肖建(1976-),男,博士,教授,CFF会员(A8837M),研究方向为嵌入式系统应用。

0 引言

随着当今社会经济和 AI 技术的快速发展, 中国移动机器人行业迎来了爆发式增长, 移动机器人也被广泛应用于工业、服务、农业、医疗等社会各领域。移动机器人通过相机等视觉传感器感知环境, 使用目标检测算法识别并定位物体。目标检测算法作为移动机器人感知环境能力中的重要一环, 在应用时也对其运行速度和精度提出了愈来愈高的要求。目标检测算法模型庞大, 占用较多的存储资源, 且计算复杂, 通常运行于算力强大, 计算资源丰富的设备上, 而嵌入式设备存储空间有限, 计算资源不足^[1]。目标检测算法在嵌入式设备中部署往往存在因模型庞大导致存储空间不足、检测算法准确度不高和运行效率低下等问题。因此, 适用于移动机器人的目标检测算法已成为当今机器人研究领域的一大热点。

目前, 目标检测算法可以大致分为两类。一类是二阶段算法模型, 如基于区域的卷积神经网络 R-CNN^[2]、Fast R-CNN^[3]、Faster R-CNN^[4]。这类二阶段算法模型首先需要获取目标候选框, 然后对候选框进行一系列的分类和回归操作, 其优点是识别精度较高。另一类是单阶段算法模型, 其中 YOLO 系列算法是最典型的代表, 其优点是识别速度快、识别精度高。虽然二阶段算法模型精度较高, 但一阶段算法模型所需计算资源更少, 运行速度更快, 更适合部署在嵌入式设备中, 因此基于 YOLO 改进的算法模型在嵌入式部署领域应用更为广泛。2020 年, 国外 Adarsh 等人提出了 YOLOv3-tiny, 该算法在 YOLOv3 的基础上去掉了一些特征层, 只保留两个独立预测分支, 在运行速度上有了较大的提升^[5]。Fang Wei 等人提出了 Tinier-YOLO, 该算法在 YOLOv3-tiny 基础上进一步缩小模型尺寸, 并提高检测精度和实时性^[6]。2021 年, 国内南京邮电大学的张伟等人人在 YOLOv3-tiny 的基础上提出了 DS-YOLO, 该网络结构简单、计算量小, 适合部署于无人机终端上^[7]。2020 年 6 月, 在 YOLOv4^[8] 算法提出的两个月后, Alexey A B 等人又提出了其轻量化网络 YOLOv4-tiny^[9]。同年, Jiang Zicong 等人人在 YOLOv4-tiny 的基础上利用 ResBlock-D 模块替换 CSPBlock 模块, 降低了计算复杂度, 并利用辅助残差网络块, 提升检测精度, 最终提出了一种高性能的实时目标检测算法^[9]。2021 年, Wang Guanbo 等人提出了 TRC-YOLO, 该算法对 YOLO v4-tiny 的卷积核进行修剪, 通过构建 CSPResNet 结构增强网络特征提取能力, 最终在 Jetson Xavier NX 上实现了每秒 36.9 帧的实时性能^[10]。2020 年, Jocher 提出 YOLOv5 模型中的 YOLOv5s, 不仅识别精度高, 而且有较高的实时性, 但在实际移动机器人应用场景中, 嵌入式设备的计算资

源仍不足。

为了更好地实现移动机器人平台下的实时目标检测, 根据 YOLOv5 在嵌入式设备部署的不足^[11], 该文提出针对性的改进方法: 一方面针对 YOLOv5s 的 backbone 网络比较复杂的问题, 重构其 backbone 网络, 并引入深度可分离卷积, 这种卷积相较于普通卷积计算量更小, 能降低模型复杂度, 提高网络实时性, 在引入深度可分离卷积后不可避免地会产生大量信息损失, 通过在 backbone 网络中引入残差结构还原损失的信息, 提升该轻量化后网络的精度; 另一方面以 YOLOv5l 作为教师网络, YOLOv5s 作为助教网络指导轻量化后的 DID-YOLO 网络训练, 因 backbone 网络被重构, 因此特征信息大量流失, 通过对该网络进行特征层和输出层蒸馏, 提升 DID-YOLO 网络精度。

1 相关工作

1.1 YOLOv5s 网络介绍

YOLO 算法将图像划分为单元格, 并在每个单元格中判断候选框。如果目标的中心落在一个单元格里, 这个单元格负责预测目标。YOLOv5 是当前主流的目标检测算法, 其分为 YOLOv5s、YOLOv5m、YOLOv5l、YOLOv5x 四个模型。这四种模型的网络结构一致, 但模块数量和卷积核数量依次递增, 其中 YOLOv5s 为轻量型网络, 广泛应用于嵌入式设备中, 具有较高的精度和实时性能。

如图 1 所示, YOLOv5s 网络分为三个部分: 第一部分是 backbone 网络, 负责目标的特征信息提取。它由 Focus 模块、BottlenCSP1 模块和 SPP 模块组成。Focus 模块将图像数据分成四部分, 每部分可等效为对输入图像进行两次下采样。通过在信道维度上进行拼接和卷积的操作来最大限度地减少因对图像下采样而造成的信息损失。BottlenCSP1 结构将 CSPNet^[12] 结构添加到残差结构中, 并将梯度变化映射到特征图中, 该结构可以减少卷积神经网络的整体计算量, 降低网络复杂程度, 而且可以有效地增强卷积神经网络的学习能力。SPP 模块首先通过固定池化操作解决了输入图片尺寸不一致问题, 并最终获得相同尺寸的输出; 其次通过空间金字塔池化操作增加了网络的整体接受域, 实现多重感受野融合。第二部分是 Neck 网络-PANet^[13], 主要是对骨干网络提取的特征信息进行增强。第三部分是 head 网络, 采用与 YOLOv3 相同的检测结构, 三个检测头分别对原始图像进行了 8 次、16 次和 32 次采样。

1.2 嵌入式平台设计方案

该文搭建的移动机器人主控为 Jetson AGX Xavier, 其结合相机组成目标检测算法验证的嵌入式

平台。Jetson AGX Xavier 作为全球首款专为自主机器打造的计算机,其板载的 Xavier Soc 具有一个 Carmel 架构 8 核 64 位的 CPU 和 Volta 架构 512 CUDA 处理器 GPU,能以低至 10W 的功耗提供 32 TOPs 的强大算

力,作为 AI 边缘计算平台,利用英伟达丰富的 AI 工具和完备的生态链资源,使得开发者能够在较短的时间内完成神经网络模型的训练和部署。

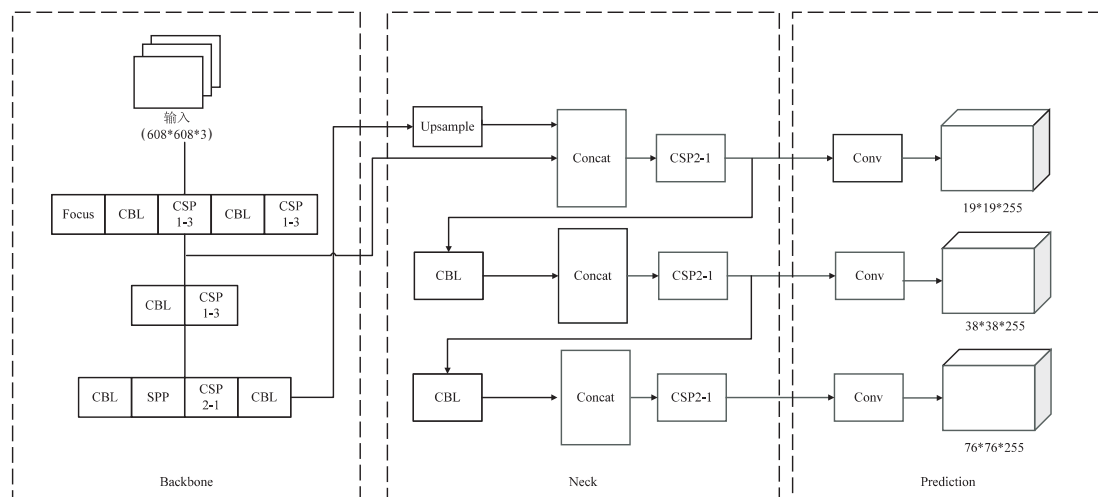


图 1 YOLOv5s 网络结构示意图

2 基于 YOLOv5s 改进的目标检测网络

2.1 轻量化网络

2.1.1 深度可分离卷积

卷积是人工神经网络中一个非常重要的数学运算。卷积神经网络(CNN)可以在图像帧的帮助下学习特征和分类数据。深度可分离卷积是 CNN 常用卷积中的一种,与标准卷积相比,它们需要调整的参数更少,且深度可分离卷积计算量较少,这使得它更适合应用于移动视觉类型的 CNN 中,其中谷歌提出的 MobileNet^[14]、Xception^[15]等网络为深度可分离卷积在移动视觉领域的典型应用。

深度可分离卷积可分为两个过程:逐通道卷积和逐点卷积。逐通道卷积:一张彩色图片可分为 r、g、b 三个通道,每个通道都与一个卷积核进行卷积可得到与输入图片通道数一致的特征图,如图 2 所示。逐点卷积:采用的是 1×1 卷积的方式,将未卷积前的图在深度上进行加权组合,生成特征图,如图 3 所示。

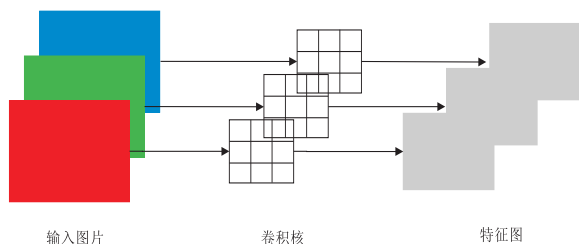


图 2 逐通道卷积示意图

由图 2 和图 3 可知,当卷积核大小为 3 时,则生成 m 个特征图所需的参数量 $pn1$ 如公式(1)所示。普通卷积生成 m 个特征图所需的参数量 $pn2$ 如公式(2)所

示。由公式(1)和公式(2)可知,当 $m > 1$ 时,生成同数量的特征图深度可分离卷积所需参数量小于普通卷积所需参数量,因此该文以深度可分离卷积重构 YOLOv5s 的 backbone 网络,降低网络复杂度。

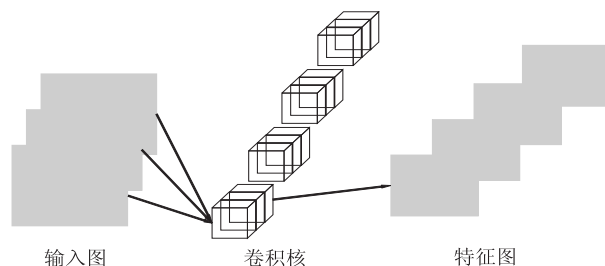


图 3 逐点卷积示意图

$$pn1 = 3 * 3 * 3 + 1 * 1 * 3 * m = 27 + 3m \quad (1)$$

$$pn2 = 3 * 3 * 3 * m = 27m \quad (2)$$

2.1.2 倒置残差结构

在引入深度可分离卷积后,网络复杂程度下降,但随着 backbone 网络模型缩小不可避免会产生大量特征信息损失。该文通过在 backbone 网络中引入倒置残差结构还原损失的信息,提升该轻量化后网络的精度。ResNet^[16]中提出的残差结构由 1×1 卷积、3×3 卷积和 1×1 卷积顺序组成,实现网络先降维后升维的操作。倒置残差结构则采用了 3×3 的深度可分离卷积并将网络先升维后降维,将高维特征映射到低维中,通过提升通道数,获得更多的特征信息。当步长为 1 时,如图 4(a)所示。首先通过 1×1 的卷积升维,扩张通道数为原来的 6 倍,然后使用 3×3 的深度可分离卷积进行特征信息提取,再通过 1×1 的卷积降维至原通道数量。当步长为 2 时,如图 4(b)所示。由于卷积的步长不一致倒置输入输出的尺寸存在差异,因此不将

输入输出进行拼接融合。

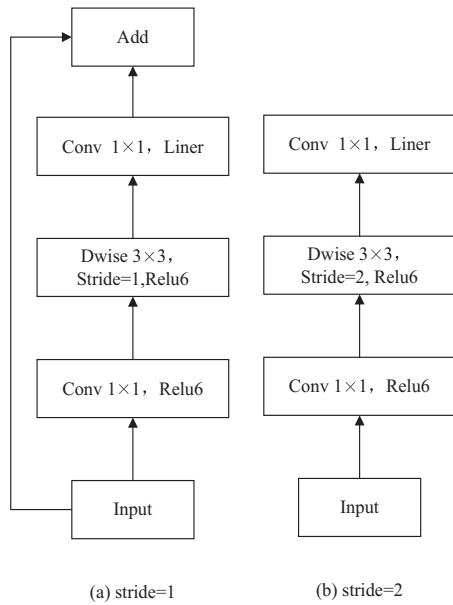


图 4 倒置残差结构示意图

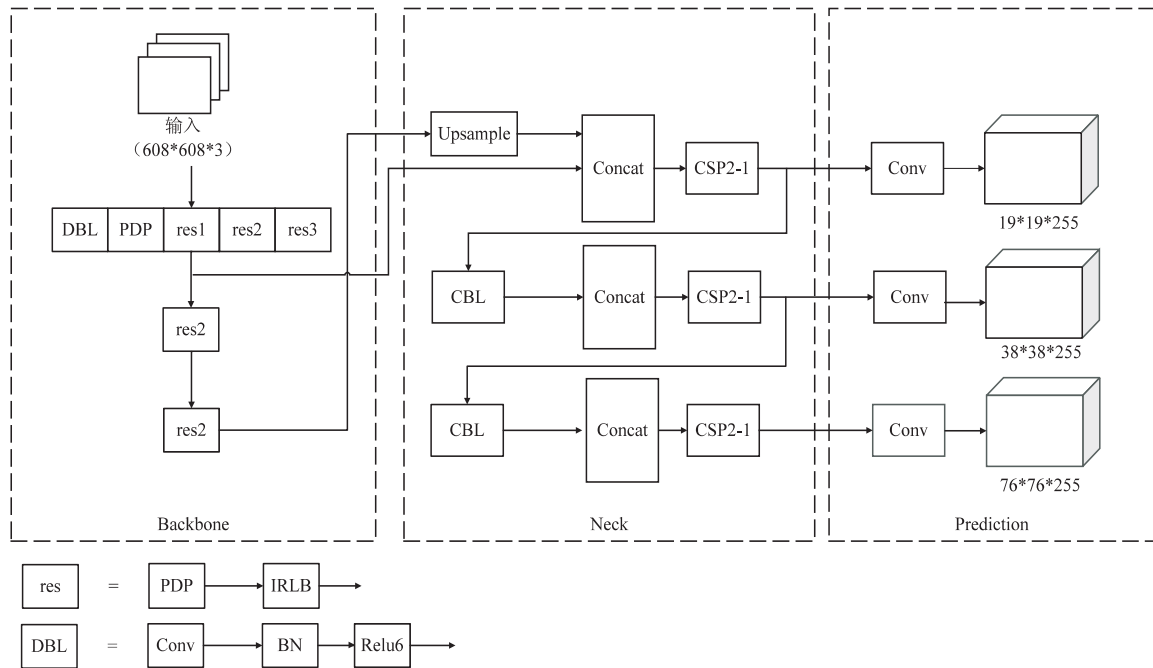


图 5 DID-YOLO 网络框架结构示意图

DID-YOLO 算法的整体流程如图 6 所示。该网络模型的总体结构由输入端处理模块、backbone 网络、neck 网络以及输出端处理模块组成。在输入端处理模块中, YOLOv5 引入了 Mosaic 数据增强的方法, 选用四张图片以随机缩放、随机裁剪、随机排布的方式将图片拼接成训练图片。Mosaic 数据增强可以丰富数据集, 在拼接过程中图片会随机缩放, 拼接完成后数据集中将增加大量小目标数据, 提升小目标的检测效果



图 6 DID-YOLO 算法流程

2.1.3 DID-YOLO 算法的建立

针对目标检测算法在嵌入式设备中运行效率低、占用存储资源过多的问题, 该文使用深度可分离卷积和倒置残差模块对 YOLOv5s 的 backbone 网络进行重构, 降低模型复杂度和计算量, 减少模型占用空间, 提高其部署在嵌入式设备时的检测速度。提出的基于 YOLOv5s 改进后适用于嵌入式设备的轻量型目标检测算法 DID-YOLO 网络结构如图 5 所示。图 5 中虚线框中是轻量化重构后的 backbone 网络, 负责提取图像中的特征信息。其中 DBL 为卷积 + BatchNormalization+relu6, PDP 为 stride=2 的倒置残差结构块, IRLB 为 stride=1 的倒置残差结构块, resn 即为 n 个 res 结构相连接。DBL、PDP 和 IRLB 作为基础模块构成了 DID-YOLO 的 backbone 网络, 使用深度可分离卷积和倒置残差模块对 YOLOv5s 的 backbone 网络进行重构后, 以 VOC 为数据集进行训练, 计算量为 4.65 G, 最终生成的模型参数数量为 3.63 M。

和网络的鲁棒性。在训练时, 针对不同数据集初始锚框值处理的问题, 输入端处理模块中引入自适应锚框计算, 每次训练时可以自适应计算出训练集的最佳初始锚框值。在推理中, 针对不同尺寸的输入图片, DID-YOLO 在输入端处理模块中引入了自适应图片缩放, 直接在原始图像中自适应地添加最少的黑边, 大大减少了推理计算量, 提高了目标检测速度。

DID-YOLO 的 backbone 网络用于提取输入端处理模块处理后图像的特征信息,经过轻量化后的 backbone 网络解决了原网络结构复杂,参数数量和计算量庞大的问题,提高了算法的运行速度。DID-YOLO 的 Neck 网络采用 PANet 结构,用于处理特征图生成特征金字塔,增强模型对不同尺寸目标对象的检测精度,达到网络高精度识别不同大小和尺寸物体的目的。PANet 结构为在 FPN 的基础上增加一个自底向上方向的增强,使得顶层特征图融合底层特征图的位置信息,从而提升对大尺寸目标物体的检测效果。

为解决预测框和真实框完全不重合时,损失函数不可求导的问题,DID-YOLO 的输出端处理模块选用 CIOU_Loss 函数作为 Bounding box 的损失函数,其计算公式为:

$$v = \frac{4}{\pi^2} \left(\arctan \frac{W^{gt}}{h^{gt}} - \arctan \frac{W^p}{h^p} \right)^2 \quad (3)$$

$$CIOU_Loss = 1 - CIOU = 1 - \left(IOU - \frac{Distance^2}{Distance_C^2} - \frac{v^2}{(1 - IOU) + v} \right) \quad (4)$$

CIOU_Loss 不但考虑到了预测框和真实框重叠的面积,还引入了预测框和真实框的长宽比和两者中心点的距离。针对多目标框的筛选,YOLOv5 的输出端处理模块选用 nms 非最大值抑制,保留检测效果最好的预测框,去除冗余的预测框。

2.2 知识蒸馏

知识蒸馏是一种常用的模型压缩方法。现有的主

流知识蒸馏方法大致可分为两类。第一类知识蒸馏,使用由教师网络生成的分类软标签和真实标签来指导学生网络训练。Jimmy Ba 等人^[17]首先提出了这类方法。但是,随着网络复杂程度不断加深,单纯依靠教师网络的软标签来蒸馏训练学生网络会产生信息沟壑,导致蒸馏训练对学生网络精度提升效果减弱。因此,第二类利用教师网络的特征层对学生网络进行指导的知识蒸馏方法被提出。Junho Yim 等人^[18]在学生网络训练中提出了一种模拟多层教师网络特征的方法。Hui Wang 等人^[19]提出了一种逐步蒸馏的方法,允许学生网络逐步学习教师网络中间层的特征信息。

该文在 YOLOv5s 基础上对其 backbone 网络进行轻量化后提出 DID-YOLO,造成 DID-YOLO 大量特征层信息损失,因此提出了针对特征图和输出层的知识蒸馏,其蒸馏框架如图 7 所示。因 backbone 网络进行重构后,其输出特征图维度与原网络输出的特征图维度不一致,采用在 DID-YOLO 的 backbone 网络输出上添加 Converter 模块的方法,使其升维至与教师网络匹配。其中 Converter 模块由 conv、bn 和 relu6 组成。因此教师网络在其 backbone 网络输出上套接 relu6 函数,保证学生和教师网络的特征层的激活函数一致。该文提出的知识蒸馏结构在输出层蒸馏的基础上,对特征提取也进行了蒸馏,提升学生网络的 backbone 在特征提取上的表征能力,在一定程度上提高了提出的基于 YOLOv5s 轻量化网络的识别精度。

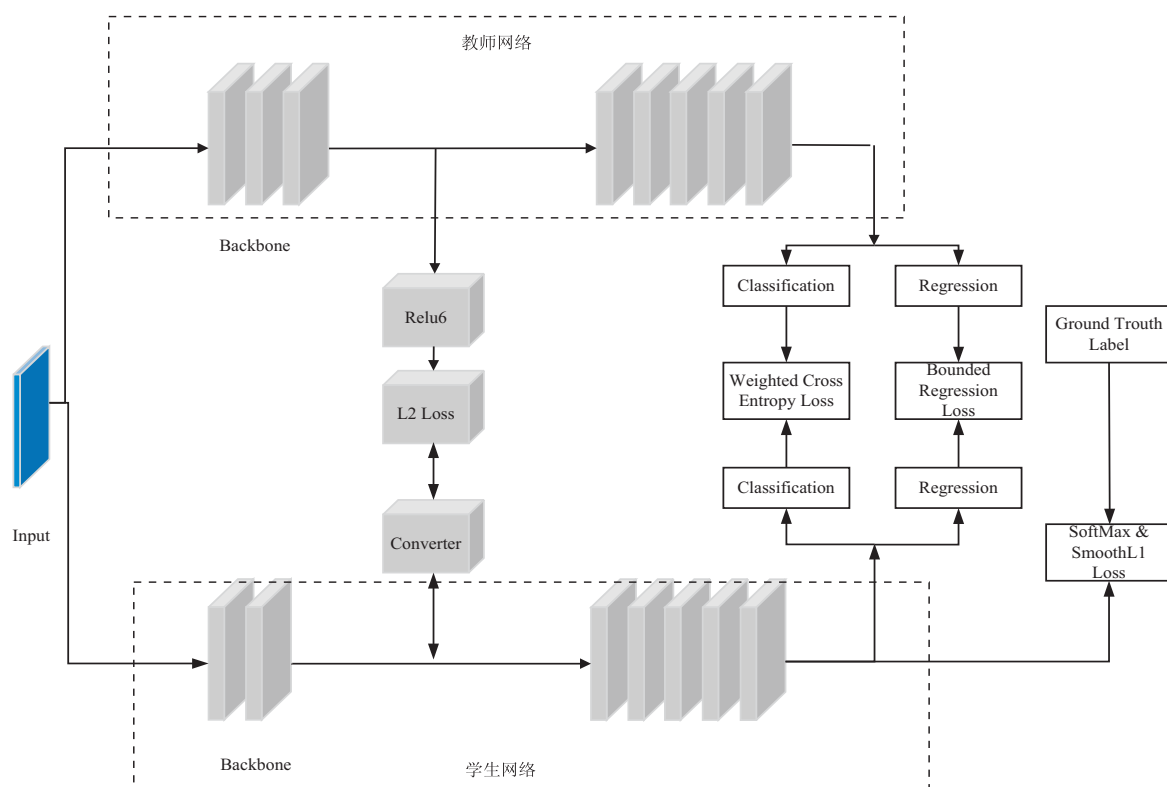


图 7 知识蒸馏框架示意图

3 实验结果与分析

3.1 数据集与实验设置

为了验证提出的移动机器人目标检测算法 DID-YOLO 的有效性, 该文所有算法模型均在公共数据集 Pascal VOC 上完成训练, 并在其验证集上验证模型的正确性。其中随机筛选 16 552 张图片作为训练集用以网络模型训练, 4 953 张图片作为验证集用以验证网络模型的正确性。提出的移动机器人目标检测算法训练和验证的环境为 ubuntu18.04 操作系统, 硬件配置为 Intel® Xeon® Silver4210 CPU @ 2.20 GHz 和 TeslaP100-PCI-E GPU, 内存为 12 GB, 总显存为 16 GB, 该算法在 cuda10.1, Pytorch1.7.0 下完成模型的搭建和网络训练。

3.2 模型评价指标

在比较模型性能时, 以平均准确率 (mAP@0.5)、模型尺寸 (Model Size)、检测速度 (fps) 衡量算法的精度、复杂程度和实时性能。其中 mAP@0.5 是指均交比阈值为 0.5 时, 所有类别的平均 AP, 其计算公式为:

$$\text{mAP@0.5} = \frac{1}{N} \sum_{i=1}^N \text{AP}_i \quad (5)$$

mAP@0.5 衡量模型精确率随召回率变化的趋势, 是目标检测算法使用 VOC 数据集测试精度最常用的评价指标。其中检测速度是由目标检测算法处理每张相机实时捕捉的图片所用的平均时间, 是评价算法运行速度的常用评价指标。

3.3 知识蒸馏效果对比

提出的移动机器人目标检测算法 DID-YOLO 在以不同网络为教师网络指导训练后的结果如表 1 所示。轻量化网络 DID-YOLO 的 mAP@0.5 为 68.05%, 以 YOLOv5l 为教师网络进行知识蒸馏后, DID-YOLO 算法的 mAP@0.5 提升至 72.24%, 提升了 6.16%; 以 YOLOv5s 为教师网络进行知识蒸馏后, DID-YOLO 算法 mAP@0.5 提升至 72.49%, 提升了 6.52%; 以 YOLOv5l 为教师网络、YOLOv5s 为助教网络进行知识蒸馏后, DID-YOLO 算法 mAP@0.5 提升至 73.83%, 提升了 8.49%。

表 1 知识蒸馏结果对比

学生网络	助教网络	教师网络	mAP@0.5/%
DID-YOLO	无	无	68.05
DID-YOLO	无	YOLOv5l	72.24
DID-YOLO	无	YOLOv5s	72.49
DID-YOLO	YOLOv5s	YOLOv5l	73.83

由表 1 可知, DID-YOLO 算法的 mAP@0.5 值最高达到 73.83%。相较于以 YOLOv5l 为教师网络的知识蒸馏训练, 引入 YOLOv5s 作为助教网络辅助蒸馏可以得到最高的 mAP@0.5 值。YOLOv5l 具有最高的

精度, 但其网络模型在结构和复杂度上与学生网络存在较大的差异, 因此引入 YOLOv5s 作为助教网络可以提高知识蒸馏能力, 辅助指导学生网络训练。

3.4 多种目标检测算法性能对比

为验证提出的移动机器人目标检测算法 DID-YOLO 部署在嵌入式设备中的效果, 在 VOC 数据集中测试了当前嵌入式平台使用的主流轻量化目标检测算法, 包括 YOLOv3-Tiny、YOLOv4-Tiny 和 YOLOv5s, 实验结果以平均精度均值 mAP@0.5、网络模型大小和在 Xavier 上运行的实时帧数为评价指标, 对比结果如表 2 所示。

表 2 多种检测模型结果对比

网络模型	mAP@0.5 /%	Model Size /MB	检测速度 /fps
YOLOv3-Tiny ^[5]	52.78	34.9	24.6
YOLOv4-Tiny ^[9]	62.57	23.6	25.1
YOLOv5s ^[20]	74.23	7.07	29.6
DID-YOLO	73.83	3.63	31.2

以 Pascal VOC 为数据集进行训练, YOLOv5s 的计算量为 8.39 G, 生成的模型参数量为 7.07 MB。经过深度可分离卷积和倒置残差模块对 YOLOv5s 的 backbone 网络进行重构后, 计算量为 4.65 G, 最终生成的模型参数量为 3.63 MB。相较于 YOLOv5s, DID-YOLO 网络模型参数量下降了 48.65%, 计算量降低了 44.58%。对比实时性能, 在以 AGX Xavier 为主控的移动机器人嵌入式平台, DID-YOLO 实时运行可达到每秒 31.2 帧的速度, 相较于 29.2 帧/s 的 YOLOv5s 提升了 5.41%。在检测精度上, DID-YOLO 最高 mAP@0.5 可达 73.83%, 相较于 YOLOv5s, 下降了 0.54%。经过轻量化后的 DID-YOLO 损失了一定程度的检测精度, 但实时性能有较大的提升。

与当前移动机器人领域中主流的目标检测算法 YOLOv3-Tiny 和 YOLOv4-Tiny 相比, DID-YOLO 在模型尺寸、检测准确率和速度上都有着较大的优势。由表 2 可知, 提出的 DID-YOLO 的模型尺寸仅为 YOLOv3-Tiny 的 10.4%、YOLOv4-Tiny 的 15.38%, 其模型尺寸被压缩至 3.63 MB, 在嵌入式设备部署时仅需占用极小的空间资源。虽然 DID-YOLO 模型尺寸较小, 但在公共数据集 Pascal VOC 上其检测准确率超过了 YOLOv3-Tiny 和 YOLOv4-Tiny, DID-YOLO 的 mAP@0.5 分别提高了 21.05% 和 11.26%。在 AGX Xavier 上, DID-YOLO 处理实时图像性能也高于两者, 其检测速度达到 31.2 fps, 相较于 YOLOv3-Tiny 和 YOLOv4-Tiny, 其检测帧率分别提高了 6.6 帧和 6.1 帧。以上分析可得, 提出的 DID-YOLO 与原网

络 YOLOv5 相比牺牲了一部分的检测精度,但模型尺寸进一步压缩且实时性能得到一定程度提升,且综合模型轻量化指标、检测精度参数和实时检测性能指标与 YOLOv3-Tiny 和 YOLOv4-Tiny 相比,提出的 DID-YOLO 更具优越性。

4 结束语

提出的 DID-YOLO 算法,在 YOLOv5s 的基础上引入深度可分离卷积和倒置残差模块重构其 backbone 网络,减少算法复杂度,并进一步压缩模型,并通过结合特征层和输出层的知识蒸馏提高 DID-YOLO 目标检测的准确率。最终在 Pascal VOC 公共数据集上进行实验表明,DID-YOLO 算法的模型尺寸压缩至 3.63 MB,mAP@0.5 达到 73.83%,实时检测速度达到 31.2 fps,该算法完全满足一个移动机器人嵌入式平台对目标检测算法的模型尺寸、检测速度和精度的要求。

参考文献:

- [1] LIU Z, FANG F, QIAN K, et al. Optimization method of target detection and tracking system for mobile robot[C]//Proceedings of the 2020 IEEE international conference on real-time computing and robotics (RCAR). Asahikawa; IEEE, 2020:410-415.
- [2] 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. Columbus; IEEE, 2014:580-587.
- [3] GIRSHICK R. Fast R-CNN[C]//Proceedings of the IEEE international conference on computer vision. Mass; IEEE, 2015:1440-1448.
- [4] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 39(6):1137-1149.
- [5] ADARSH P, RATHI P, KUMAR M. YOLO v3-Tiny: object detection and recognition using one stage improved model[C]//Proceedings of the 2020 6th international conference on advanced computing and communication systems (ICACCS). Coimbatore; IEEE, 2020:687-694.
- [6] FANG W, WANG L, REN P J I A. Tinier-YOLO: a real-time object detection method for constrained environments[J]. IEEE Access, 2020, 8:1935-1944.
- [7] 张伟, 庄幸涛, 王雪力, 等. DS-YOLO: 一种部署在无人
- 机终端上的小目标实时检测算法[J]. 南京邮电大学学报, 2021, 41(1):86-98.
- [8] BOCHKOVSKIY A, WANG C Y, LIAO H M. Yolov4: optimal speed and accuracy of object detection[J]. arXiv:2004.10934, 2020.
- [9] JIANG Z, ZHAO L, LI S, et al. Real-time object detection method based on improved YOLOv4-tiny[J]. arXiv:2011.04244, 2020.
- [10] WANG G, DING H, YANG Z, et al. TRC-YOLO: a real-time detection method for lightweight targets based on mobile devices[J]. IET Computer Vision, 2022, 16(2):126-142.
- [11] 谭任, 唐忠, 王鸿亮, 等. 基于深度学习的嵌入式汽车内饰件装配检测[J]. 计算机系统应用, 2022, 31(4):110-116.
- [12] WANG C Y, LIAO H Y M, WU Y H, et al. CSPNet: a new backbone that can enhance learning capability of CNN[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW). Seattle; IEEE, 2020:1571-1580.
- [13] LIU S, QI L, QIN H, et al. Path aggregation network for instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). Salt Lake City; IEEE, 2018:8759-8768.
- [14] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications[J]. arXiv:1704.04861, 2017.
- [15] CHOLLET F. Xception: deep learning with depthwise separable convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). Honolulu; IEEE, 2017:1800-1807.
- [16] TARG S, ALMEIDA D, LYMAN K. Resnet in resnet: generalizing residual architectures[J]. arXiv:1603.08029, 2016.
- [17] BA J, CARUANA R. Do deep nets really need to be deep[C]//Neural information processing systems. [s. l.]: MIT Press, 2014:2654-2662.
- [18] YIM J, JOO D, BAE J, et al. A gift from knowledge distillation: fast optimization, network minimization and transfer learning[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). Honolulu; IEEE, 2017:7130-7138.
- [19] HUI W, ZHAO H, XI L, et al. Progressive blockwise knowledge distillation for neural network acceleration[C]//Twenty-seventh international joint conference on artificial intelligence (IJCAI-18). Stockholm; FAIM, 2018:2770-2775.
- [20] JOCHER G. YOLOv5[EB/OL]. 2020. <https://github.com/ultralytics/yolov5>.