控制与决策
Control and Decision

基于ResNet34_D改进YOLOv3模型的行人检测算法

钱惠敏, 陈纬, 马宜龙, 施非, 项文波

引用本文:
钱惠敏,陈纬,马宜龙,施非,项文波. 基于ResNet34_D改进YOLOv3模型的行人检测算法[J]. 控制与决策, 2022, 37(7): 1713-1720.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2021.0136

基于ResNet34_D改进YOLOv3模型的行人检测算法

钱惠敏[1†], 陈 纬[1], 马宜龙[1], 施 非[1], 项文波[2]

(1. 河海大学 能源与电气学院，南京 211100；2. 南京理工大学 自动化学院，南京 210094)

摘 要: 针对自动驾驶场景下行人检测任务中对中、小尺寸目标和被遮挡目标的检测需求,以及现有深度学习模型的不足,提出基于ResNet34_D的改进YOLOv3模型:通过改进残差网络的卷积块结构提出ResNet34_D,并作为YOLOv3的主干网络以降低模型尺寸和训练难度;在ResNet34_D的3个尺度卷积特征图之后,增加SPP层和DropBlock模块以提高模型的泛化能力;基于K-means聚类算法确定自适应的多尺度锚框尺寸,提高对大、中、小3种尺寸行人目标的检测能力;引入DIoU损失函数,提高对被遮挡目标的识别能力. 所提出模型的消融实验验证了各个改进部分在提高模型检测准确率上的有效性. 实验结果表明,所提出的基于ResNet34_D的改进YOLOv3模型具有较好的准确率和实时性,在BDD100K-Person数据集上的AP_{50}达到69.8 %,检测速度达到130 FPS. 由所提出方法与现有目标检测方法的对比实验可知,所提出方法对小目标和遮挡目标的误检率更低,速度更快,具有一定的实际应用价值.

关键词: 行人检测；深度学习；YOLOv3；ResNet34_D；DIoU

中图分类号: TP391 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.0136

引用格式: 钱惠敏,陈纬,马宜龙,等. 基于ResNet34_D改进YOLOv3模型的行人检测算法[J]. 控制与决策, 2022, 37(7): 1713-1720.

Pedestrian detection based on developed YOLOv3 with ResNet34_D

QIAN Hui-min[1†], *CHEN Wei*[1], *MA Yi-long*[1], *SHI Fei*[1], *XIANG Wen-bo*[2]

(1. College of Energy and Electrical Engineering，Hohai University，Nanjing 211100，China；2. College of Automation，Nanjing University of Science and Technology，Nanjing 210094，China)

Abstract: Pedestrian detection is one of the main tasks of autonomous driving. The existed deep neural network is lack of the ability to detect small-size or medium-size objects and occluded objects, which is the requirement of pedestrian detection since pedestrians in the images acquired by vehicle-equipped cameras are always small or medium or occluded. In this paper, an improved YOLOv3 model based on ResNet34_D is proposed for pedestrian detection. And the contributions of the improved model are as follows. Firstly, the developed residual network ResNet34_D by modifying the structure of convolutional block is proposed, and it is selected as the backbone of YOLOv3 to reduce the size of the model so as to decrease the training difficulty. Secondly, the SPP layer and the DropBlock module are introduced after the feature maps of three stages of ResNet34_D, which can improve the detection accuracy of pedestrian objects with different sizes. Thirdly, to further increase the detection accuracy, the multi-scale anchors are determined using the K-means. Finally, the DIoU loss function is used to improve the ability of detecting the occluded objects. Ablation experiments for the proposed model demonstrate the effectiveness of each developed technologies in improving detection accuracy. And more experimental results show that the AP_{50} of the proposed model on BDD100K-Person dataset reaches 69.8%, and the detection speed can achieve 130 FPS. Comparison experiments between the proposed method and the other existed methods demonstrate that, using the proposed method, the false detection rate for small targets and occlusion targets is lower, and the speed is faster, therefore, the proposed improved YOLOv3 model based on Resnet34_D is valuable in practical applications.

Keywords: pedestrian detection；deep learning；YOLOv3；ResNet34_D；DIoU

收稿日期: 2021-01-22；**录用日期:** 2021-04-21.

基金项目: 中央高校基本科研业务费专项基金项目(2018B15514).

责任编委: 张国山.

†通讯作者. E-mail: qhmin0316@163.com.

0 引 言

基于深度卷积神经网络的计算机视觉技术已取得了举世瞩目的成就,这些成就推动了无人驾驶技术的发展. 行人检测是无人驾驶技术中的一项重要研究内容,也是计算机视觉领域的研究热点之一[1]. 近几十年,研究人员在行人检测方面取得了一系列的研究成果. 目前,常用的行人检测方法主要分为基于人工设计特征和浅层机器学习模型的传统方法以及基于深层机器学习模型的方法[2].

传统的行人检测方法主要采用基于人工设计的特征提取规则,并采用浅层机器学习模型实现行人检测. 但是,传统方法由于泛化能力不足、检测速度慢、难以适应复杂的行人姿态,从而难以满足实际应用需求.

随着深度神经网络,特别是卷积神经网络在目标检测任务中的发展,以及大规模行人检测公共数据集的出现,基于深度神经网络的行人检测算法已在实际问题中得到了应用. 基于深度神经网络的目标检测算法包括二阶段目标检测算法和一阶段目标检测算法. 二阶段目标检测算法通常包含候选区域生成和分类回归两部分,其代表性算法是区域卷积神经网络(region-convolutional neural networks, R-CNN)[3]及其改进的Fast R-CNN[4]、Faster R-CNN[5]、Mask R-CNN[6]、Cascade R-CNN[7]等. 一阶段目标检测算法无需候选区域生成部分,而是直接通过回归来预测检测框,将检测转化为回归问题. 一阶段目标检测算法的代表性算法有YOLO(you only look once)[8]、YOLO9000[9]、SSD(single shot detector)[10]、YOLOv3[11]等. 一阶段和二阶段目标检测算法在通用目标检测任务中均取得了较好的检测效果. 本文研究的行人检测问题,属于特定目标检测任务. 已有一些研究工作将这两类检测算法引入行人检测问题. 例如,陈泽等[12]提出了一种基于改进Faster R-CNN的目标检测方法,通过引入基于双线性插值的对齐池化层,并通过设计基于级联的多层特征融合策略,较好地解决了小尺度行人在深层特征图中特征信息缺乏的问题. 黄同愿等[13]提出了精简YOLOv3的主干网络,以及修改锚尺寸和损失函数的方法,在BDD100K-Person数据集上的检测精度达到了53.7%,检测速度达到130 FPS.

虽然现有一阶段、二阶段目标检测算法在行人检测问题中已取得了一些研究进展,但仍然存在小尺寸行人目标和被遮挡行人目标被误检、漏检或重复检测的问题. 此外,在自动驾驶应用场景下,行人目标检测算法应同时具备实时性和准确性,对于任何复杂场景和任意多的目标都能及时作出准确的响应[13]. R-CNN和Faster R-CNN等二阶段方法由于实时性难以得到保证而较难实现应用,而YOLOv2和YOLOv3等一阶段算法在经过不断地改进后,在检测的实时性方面性能十分突出,但在部署方面依然存在一定的困难.

针对以上问题,本文采用多种手段改进适用于行人检测的YOLOv3模型,包括: 1) 调整网络结构,通过修改残差网络的卷积块结构,提出ResNet34_D作为YOLOv3的主干网络,相比于DarkNet53, ResNet34_D拥有更少的卷积层、更快的检测速度以及较好的检测精度,满足行人检测的实际需求且易于部署在低性能的设备上; 2) 在ResNet34_D的3个尺度卷积特征图之后,增加空间金字塔池化(dpatial pyramid pooling, SPP[14])和DropBlock[15]模块,以提高模型的泛化能力; 3) 为提高检测网络对于多尺寸目标和被遮挡目标的检测能力,基于K-means聚类算法确定自适应的多尺度锚框尺寸,并引入DIoU损失函数代替原有边框位置损失函数,充分考虑候选框与真实框之间重叠比、中心点距离以及长宽比.

1 YOLOv3算法的基本原理

YOLOv3算法的基本原理是将固定大小的图像作为网络的输入,基于回归获得边界框的位置及其所属类别,从而实现端到端的目标检测. YOLOv3的主干网络Darknet-53由52个卷积层和1个全连接层组成,激活函数为Leaky ReLU. 此外,Darknet-53还借鉴了ResNet[16]残差网络的近道连接的思想.

YOLOv3在应对因目标尺寸变化而引起识别困难方面,采用两种方法: 1) 使用3种类型降采样,分别为32倍、16倍和8倍降采样,从而输出3个不同尺度特征图,即52×52、26×26和13×13,通道数为255; 2) 借鉴特征金字塔网络的思想,采用多尺度融合的方法对DarkNet-53主干网络输出的3个不同尺度的特征图进行特征融合. 多尺度融合的方法可以获得更好的细粒度特征及更有意义的语义信息,同时,在训练过程中随机改变输入图像大小,可以实现网络模型的多尺度训练,从而提升算法对小目标的敏感度与检测精度.

YOLOv3的损失函数由边界框位置信息损失$L_{\rm pos}$、边界框置信度得分损失$L_{\rm conf}$以及相应类别概率得分损失$L_{\rm cl}$组成,即

$$L=\frac{1}{2}\sum_{i=1}^{M}\lambda_{\rm obj}\times(L_{\rm pos}+L_{\rm cl})+L_{\rm conf}. \tag{1}$$

其中

$$L_{\rm pos}=(2-T_w\times T_h)\times\sum_{r\in(x,y,w,h)}(T_r-P_r)^2,$$

$$L_{\rm conf}=(T_{\rm conf}-P_{\rm conf})^2,$$

$$L_{\rm cl}=\sum_{r=0}^{K-1}(I_r-P_{{\rm cl}_r})^2.$$

这里: M 表示一张图片中的样本总数; $\lambda_{\rm obj}$ 表示区域中是否含有目标,当图像中有目标时取1,否则取0; x、y 分别为目标区域中心的横、纵坐标; w、h 分别为目标区域的宽度、高度; T 代表真实值, P 代表预测值; cl表示类别, K 表示类别数; I_r 表示 r 类别是否为真实目标类别,当 r 为真实目标类别时为1,否则为0.

2　基于ResNet34_D的改进YOLOv3行人检测算法

行人检测任务的规模和难度与多类别目标检测任务截然不同,将用于多类别目标检测任务的YOLOv3算法直接应用于行人检测任务,会面临以下问题: 模型的主干网络太深,易造成资源浪费; 没有重点考虑行人检测中常出现的小目标、遮挡目标等难以检测的问题. 因此,本文在文献[11]的基础上设计用于特定目标检测的行人检测器,使得改进后的网络在行人检测任务中获得最佳性能,具体设计过程如下.

1) 通过修改残差网络ResNet34的卷积块结构,提出ResNet34_D网络,并作为YOLOv3的主干网络,从而降低模型尺寸,降低模型的训练难度;

2) 在ResNet34_D的3个尺度卷积特征图之后,增加SPP层和DropBlock模块,提高模型的特征表达能力和泛化能力,特别是对被遮挡目标的检测能力;

3) 针对数据集的分布特性,对数据集的目标真实框尺寸进行 K-means聚类,确定自适应的多尺度锚框尺寸,提高对大、中、小3种尺寸行人目标的检测能力;

4) 充分考虑候选框与真实框之间重叠比、中心点距离以及长宽比,引入坐标误差损失函数DIoU,在不增加模型大小以及推理速度的情况下,提升行人检测精度.

本文提出的基于ResNet34_D的改进YOLOv3网络结构如图1所示. 其中: 网络输入图像尺寸为608 × 608 × 3, 3个不同尺度的特征图输出 y_1、y_2 和 y_3 的尺寸分别为19 × 19、38 × 38和76 × 76. 输出多尺度特征图可应对不同环境下的行人的小、中、大3种尺度,从而提高算法对遮挡目标检测的能力,提升算法精度. 下面对网络中的各个模块给出简要介绍.

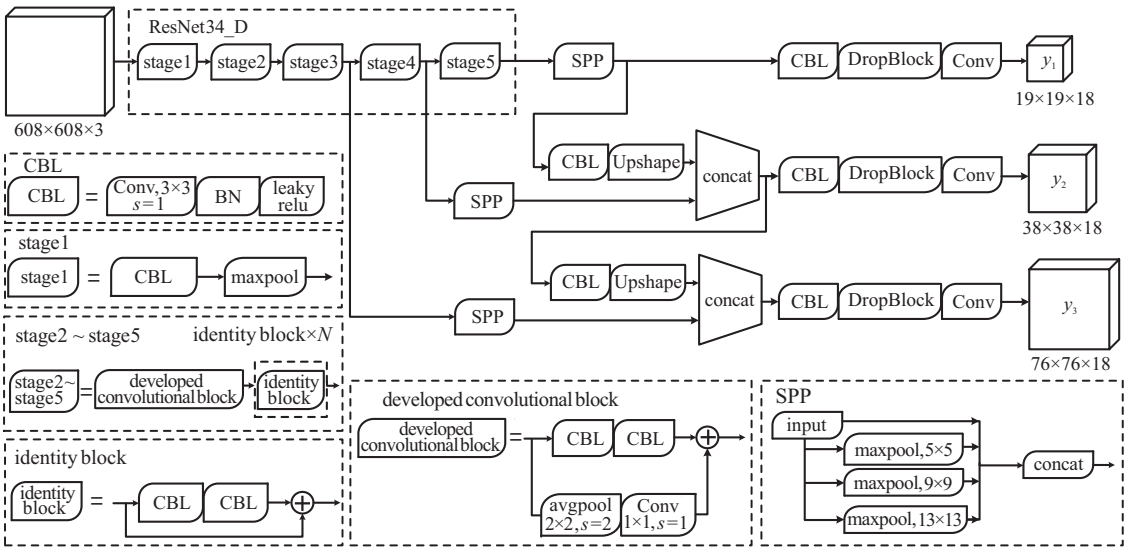


图1　基于ResNet34_D网络的YOLOv3行人检测模型结构

2.1　主干网络ResNet34_D

深层卷积神经网络的卷积层数更深,能提取更具表达力的特征图,但是,深层网络的训练更加困难. 鉴于以上问题,本文选取ResNet34作为YOLOv3的主干网络. ResNet34包含5个stage,即5种参数不同的卷积阶段. 其中: stage1阶段由一个卷积层和一个最大池化层组成,而stage 2 ~ stage 5阶段则由两种不同的残差块堆叠而成. 在YOLOv3中,这两种残差块分别是卷积块(convolutional block)和恒等块(identity block). 其中,卷积块在近道连接上引入的卷积层(卷积核大小为1×1,步长为2),将输入尺寸缩小到1/4,使得3/4的信息被丢失. 受到文献[17]的启发,本文提出改进的卷积块结构(developed convolutional block),如图1所示,在近道连接上使用了一个平均池化层(卷

积核大小为2×2,步长为2)和一个卷积层(卷积核大小为1×1、步长为1). 在卷积层之前引入平均池化层,可以保留感受野的范围,避免过多的信息丢失,又可以达到下采样的目的. 采用卷积核大小为1×1、步长为1的卷积层,使得输出特征图的尺寸与另一条路径上的输出特征图的尺寸一致,便于后续执行像素级的加法运算. 本文将改进后的卷积块应用于ResNet34网络中,得到ResNet34_D网络.

2.2 空间金字塔池化

YOLOv3网络要求输入图像固定尺寸,而为满足此要求采用的图像伸缩(剪切)等操作会造成一定程度的图像失真. 在行人检测问题中,这一要求会使得网络对遮挡目标和小目标的漏检率较高. 空间金字塔池化,使用固定分块的池化操作,对不同尺寸的输入实现相同大小的输出,可解决输入图像尺寸不统一的问题;使用多种尺寸的池化操作,可扩大特征图对应的感受野,从而应对多尺度目标表示的困难. SPP的结构如图1所示:首先,通过5×5、9×9和13×13三个大小不同的最大池化层对特征图分别进行池化,从不同大小的感受野提取特征;然后,将3种规格的池化层输出与输入特征图进行拼接,得到融合的特征向量.

针对行人检测问题中,大、中、小3个尺寸的行人目标均常见的情况,本文提出在YOLOv3网络的stage 3、stage 4、stage 5的输出特征图之后均引入SPP层,以应对行人目标的尺寸差异引起的困难,如图1所示.

2.3 DropBlock

目前,提高网络泛化能力、减少过拟合的常用方法是Dropout正则化[18]. Dropout主要通过随机删除神经单元及其连接来提高网络的泛化性能,该方法用于全连接层时取得了较好的效果,但用于卷积层时的效果欠佳. 由于卷积层中的神经单元学习的特征在空间上具有相关性,当采用Dropout技术随机删除神经单元及其连接时,这些神经单元的信息仍然能够被其周围领域内的神经单元学习.

文献[15]提出了DropBlock正则化方法,不同于Dropout删除独立的随机单元,DropBlock随机删除一部分相邻的整片区域神经单元,保留下来的神经元集中学习目标的其他部分的特征表示,从而提高网络的泛化能力,提高对被遮挡目标的识别能力. DropBlock主要有两个参数:s_b和γ. 其中:s_b表示进行归零的方块的大小,当$s_b=3$时,删除的方块大小为3×3;γ表示删除激活单元的个数,用来控制每个特征图中有多少个通道要进行DropBlock. γ参数由下式确定:

$$\gamma=\frac{1-p_k}{s_b^2}\times\frac{s_f^2}{(s_f-s_b+1)^2}. \tag{2}$$

其中:s_f^2为输入特征图的大小;$(s_f-s_b+1)^2$为经DropBlock操作后有效区域的大小;p_k为每个激活单元被保留的概率,即整个特征图中被保留的像素个数所占的比例,该值的选择会影响网络的检测准确率. 在训练过程中发现,使用固定p_k的DropBlock方法效果欠佳. 为避免此问题,本文使用一种线性降低p_k参数的方案,即随着训练步数的增加,p_k从1 逐渐减小到0.85.

2.4 确定锚框尺寸

YOLOv3算法使用先验框预测目标边界框,先验框的确定采用锚框机制,而锚的个数及宽高比是目标识别精度的影响因素之一. 文献[19]统计了COCO数据集上的目标尺寸,确定了最优的锚个数及宽高比. COCO数据集共有约80类目标数据,包含人、车、草地等,其目标以“扁长型”居多. 但是,本文实验数据集BDD 100 K中的行人目标多为“瘦高型”,原有锚框的尺寸不适用于行人检测问题. 因此,本文以BDD 100 K数据集为统计对象,对YOLOv3网络的3种尺度下的输出特征图(即图1中的y_1、y_2、y_3)采用K-means聚类算法获得适合高密度人群数据集的最优锚框个数及宽高比. 由于数据集中的行人目标大致可分为大、中、小3种目标尺寸,本文选择聚类类别数为$K=3$,而聚类的距离度量则采用候选框与真实框之间的交并比.

本文通过对BDD100K行人数据集进行聚类分析,得到在3种尺寸的特征图上对大、中、小3类人体目标的9个锚框尺寸,如表1所示.

表1 基于K-means聚类的锚框尺寸

特征图尺寸	小目标	中目标	大目标
19×19	[34, 144]	[50, 222]	[93, 317]
38×38	[11, 56]	[17, 72]	[22, 107]
76×76	[5, 19]	[7, 36]	[12, 33]

从表1中数据可以看出,最小的19×19的特征图,其感受野最大,适合检测较大的目标,因此使用较大的锚框尺寸[34, 144]、[50, 222]和[93, 317];中等的38×38特征图,因其具有中等感受野,故应用中等的锚框尺寸[11, 56]、[17, 72]和[22, 107],适合检测中等大小的目标;而较大的76×76特征图则因其具有较小的感受野,故应用最小的锚框尺寸[5, 19]、[7, 36]和[12, 33],适合检测较小的目标. 由此通过结合不同细粒度特征,增强网络对多尺度行人检测的鲁棒性.

2.5 改进的损失函数

在YOLOv3损失函数中,利用均方误差作为目标框中心坐标和宽高的损失函数,如式1所示. 由于边界框的位置信息是独立预测的,坐标之间没有明确的关联性,均方误差损失函数无法区分候选框与真实框之间不同的包含情况,不足以代表整个边界框的位置好坏. 通过引入候选框与真实框之间的面积的交并比(intersect over union, IoU)损失函数[20]可以较好地解决这一问题. IoU损失如下式所示:

$$L_{\mathrm{IoU}} = 1 - \frac{|B \bigcap B^{\mathrm{gt}}|}{|B \bigcup B^{\mathrm{gt}}|}. \tag{3}$$

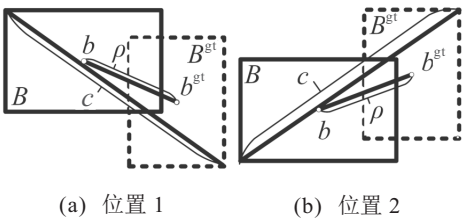
其中: $B \bigcap B^{\mathrm{gt}}$ 表示候选框 B 与真实框 B^{gt} 相交的面积; $B \bigcup B^{\mathrm{gt}}$ 表示候选框的面积加上真实框面积减去两个框相交的面积,也称之为并集. 当候选框与真实框的面积重叠越多时,IoU值越大,定位效果越好,能更好地从整体反映边界框的定位精度. 但是,IoU损失函数也存在如下问题: 当IoU的值为0时,表示候选框与真实框完全没有重叠,此时损失函数梯度为0,候选框不移动; 其次,IoU仅采用面积的交并比,忽略了候选框与真实框是如何相交的(包括中心点距离、角度、长宽比等问题),当拥有相同的IoU,而两个目标框有着不同的位置关系时,定位的效果各不相同,无法看出优化方向,从而影响检测效果.

本文引入DIoU (distance IoU loss)损失函数[21]作为位置误差损失函数,能够更好地优化边框的位置信息. DIoU损失函数在IoU损失函数的基础上充分考虑了候选框与真实框之间的重叠比、中心点距离以及长宽比,即

$$L_{\mathrm{DIoU}} = 1 - \frac{|B \bigcap B^{\mathrm{gt}}|}{|B \bigcup B^{\mathrm{gt}}|} + \frac{\rho^2(b, b^{\mathrm{gt}})}{c^2}. \tag{4}$$

其中: ρ 表示候选框中心点 b 与真实框中心点 b^{gt} 之间的欧氏距离; c 表示能够同时覆盖候选框与真实框的最小矩形的对角线距离.

DIoU通过惩罚项优化候选框与真实框之间的直接距离,如图2所示. 对于候选框与真实框的IoU相同而位置不同的情况,DIoU损失将推动候选框向真实框中心点不断靠近,有效地规避了IoU损失存在的收敛速度慢、特定情况下回归精度低等问题,在不增加模型大小以及推理速度的情况下,提升了行人检测精度.



(a) 位置 1　　(b) 位置 2

图 2　DIoU示意(候选框与真实框的不同相对位置)

3 实验及结果分析

3.1 行人检测数据集

本文以自动驾驶为背景,研究行人检测问题. 目前常用的目标检测公共数据集VOC、COCO等中虽然包含了行人目标,但包含行人目标的图像并不符合本文的研究背景. 伯克利大学AI实验室发布的BDD100K[22]数据集,是目前规模最大、内容最具多样性的自动驾驶数据集,包含12万张尺寸为1 280 × 720的高清图片. 与KITTI[23]和CityPersons[24]等其他自动驾驶场景下的行人数据集对比,BDD100 K数据集具有规模大、种类齐全、真实道路采集、拥有时间信息等优点,它包含城市和郊区道路环境,白天黑夜等自动驾驶可能遇到的各类场景.

本文选用BDD100 K数据集中带有Person标签的图片研究行人检测问题,共计25 324张,其中训练集22 032张,验证集3 292张. 需要说明的是,训练图像集中的图像将通过图像缩放、灰度填补、均值和方差标准化等图像处理方法实现训练集的扩充. 此外,在训练过程中,采用数据增强技术,对图像执行翻转、平移、旋转、颜色变换、混合图像和随机多尺度等操作,进一步扩充数据集,提高网络的泛化能力.

3.2 实验配置与训练

本文实验使用的Linux系统版本是Ubuntu16.04,服务器的硬件配置如下: CPU为Intel-Xeon-E5-2680v32.5 GHz, 内存为32 G, GPU为Nvidia RTX 2080Ti, 显存为11 G. 实验使用PaddlePaddle深度学习框架对模型进行搭建、训练和测试,CUDA版本为10.1, CuDNN版本为7.5. 在实验中所使用的Python库为Anaconda3, Python版本为3.7.

训练时,一个批次包含8幅608 × 608的图像,动量参数为0.9,采用异步梯度下降进行优化; 最大迭代次数为250 000,衰减系数为0.000 5. 学习率采用分布策略,初始学习率为0.001,迭代到150 000、200 000

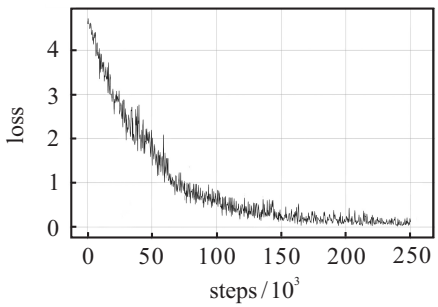


图 3　网络迭代损失值变化曲线

和240 000次时,学习率分别变为0.000 1、0.000 01和0.000 001. 改进的YOLOv3模型的迭代损失值变化如图3所示. 由图3可见,当迭代超过200 000次以后,损失值趋于稳定,最终下降到0.1左右.

3.3 实验结果与分析

3.3.1 本文提出的改进模型的消融实验

本文提出的改进YOLOv3行人检测算法包含5个改进,分别是:采用优化后的ResNet34_D作为主干网络,引入多个SPP层,引入DropBlock模块,采用K-means聚类算法确定不同尺度特征图的锚尺寸,以及使用DIoU损失函数代替边框位置损失函数. 为了验证本文改进算法的有效性,在BDD100K-Person数据集上进行各类方法的消融实验,实验结果如表2所示. 其中:AP_{50}、AR_{50}分别表示当IoU $=$ 0.5时的平均检测精度和平均召回率;FPS表示网络模型每秒钟检测的图片数量;模型大小给出了网络模型的参数量. 由表2可知,引入ResNet34_D主干网络的模型C,相比基于Darknet53主干网络的模型A,模型收敛得更快,在检测速度上增加了一倍多,达到了136.60 FPS,而平均检测精度和召回率能基本保持. 模型D、E、F、G分别是在C的基础上递进增加SPP层、DropBlock技术、基于K-means确定锚框尺寸、采用DIoU Loss,从检测精度和收敛速度来看,各项改进均能提高平均检测精度和召回率,加速了模型的收敛.

表2 在BDD100K-Person数据集上的消融实验结果

模型	YOLOv3及其改进	AP_{50} (%)	AR_{50} (%)	FPS	模型大小 (MB)
A	Darknet53	66.70	61.71	62.30	246.30
B	A+ResNet34	64.10	62.30	144.10	112.50
C	B+ResNet34_D	66.12	62.75	136.60	112.60
D	C+SPP	66.83	63.02	1132.60	112.70
E	D+DropBlock	67.22	63.93	131.90	112.80
F	E+锚框	68.96	64.41	131.70	112.80
G	F+DIoU Loss	69.80	65.40	130.00	112.50

综上,本文所提出的基于ResNet34_D的改进YOLOv3模型G,在BDD100K-Person数据集上的AP_{50}达到了69.8%,检测速度为130 FPS;与基于Darknet53的YOLOv3模型A相比,AP_{50}提高了2.1%,AR_{50}提高了3.69%,检测速度提高了一倍.

3.3.2 本文方法与现有方法的对比

为了说明本文算法的有效性,进一步将所提出算法与二阶段目标检测算法和一阶段目标检测算法中的代表性算法进行比较. 二阶段算法选择Fast R-CNN[4]、Cascade R-CNN[7],其中Cascade R-CNN算法是目前二阶段目标检测方法中检测准确率最高的算法之一. 一阶段算法选择YOLOv3及其改进算法,包括YOLOv3-SPP[25]、YOLOv3-Anchor[26]和YOLOv3-IoU Loss[27],以及YOLOv4[28]. 本文采用相同的实验配置,在BDD100K-Person数据集上实现了这些算法,结果见表3.

表3 不同目标检测算法对比

模型	算法	AP_{50} (%)	AR_{50} (%)	FPS	模型大小 (MB)
1	Fast R-CNN[4]	50.04	53.10	15.20	301.05
2	Cascade R-CNN[7]	69.61	65.10	22.10	357.05
3	YOLOv3[11]	66.70	61.71	62.30	246.30
4	YOLOv3-SPP[25]	67.60	62.21	61.10	248.30
5	YOLOv3-Anchor[26]	67.50	62.09	62.30	248.10
6	YOLOv3-IoULoss[27]	67.15	62.15	62.40	248.20
7	YOLOv4[28]	73.13	70.10	56.12	266.30
8	本文	69.80	65.40	130.00	112.50

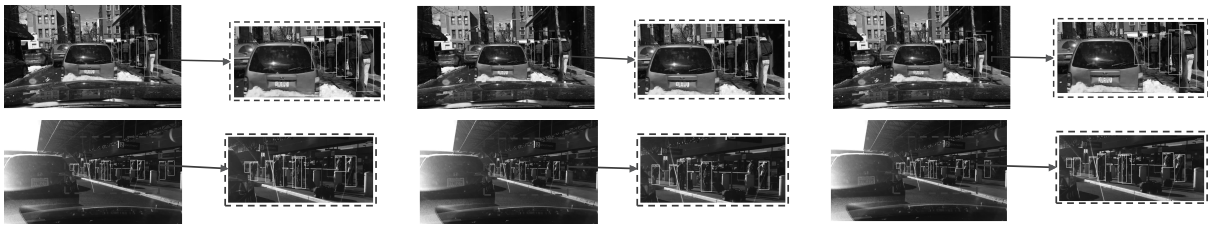
由表3可知,本文算法的检测精度和召回率均高于Fast R-CNN、Cascade R-CNN和YOLOv3算法及其改进. 但是,与YOLOv4算法相比,本文提出的算法在精度和召回率上有些许差距. 但是,本文算法的模型大小仅为YOLOv4的一半,检测速度为YOLOv4的一倍以上. 由于本文所提出的行人检测算法旨在辅助实现自动驾驶,部署设备通常是低性能设备,而模型尺寸小、检测速度快的模型则更具优势.

3.3.3 本文方法的图像检测结果及其分析

图4给出了上述3种代表性网络模型在BDD100 K验证集上的检测结果对比,检测结果的置信度阈值设置为0.5,所用算法从左向右依次为Cascade R-CNN、YOLOv3-DarkNet53、本文算法. 由图4可知,本文算法在BDD100K-Person数据集上对遮挡目标、密集小目标和模糊目标的检测效果更佳.

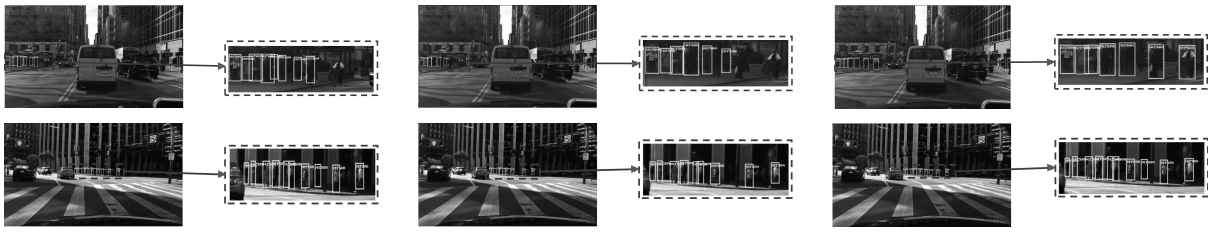
4 结 论

行人检测是无人驾驶技术中的一项重要研究,虽然现有目标检测算法在行人检测问题中已取得了一些研究进展,但仍然存在小尺寸行人目标和被遮挡行人目标被误检、漏检或重复检测的问题. 此外,在无人驾驶应用场景下,行人目标检测算法应当同时具备实时性和准确性. 针对现有技术的不足,本文提出了基于ResNet34_D的改进YOLOv3目标检测算法,通过修改YOLOv3的骨干网络、引入SPP层和DropBlock模块、基于K-means的自适应锚框尺寸,以及引入DIoU损失函数等手段,改进了目标检测模型. 本文提出的模型在公共数据集BDD100 K上的实验结果,以及与已有模型的比较,表明了本文模型的检测准确率



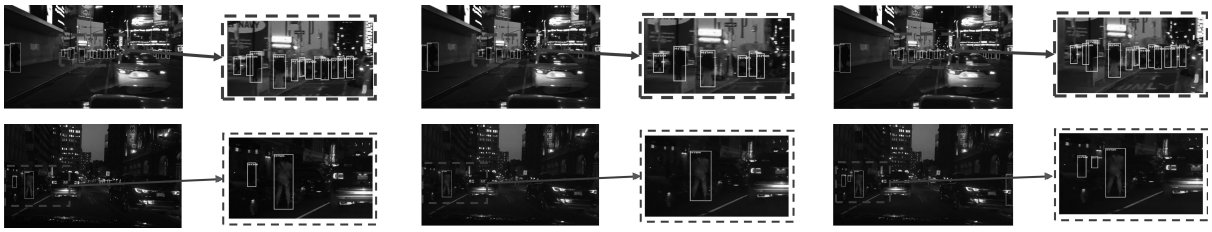
(a1) Cascade R-CNN　　(a2) YOLOv3-DarkNet 53　　(a3) 本文算法

(a) 遮挡目标的检测



(b1) Cascade R-CNN　　(b2) YOLOv3-DarkNet 53　　(b3) 本文算法

(b) 密集小目标的检测



(c1) Cascade R-CNN　　(c2) YOLOv3-DarkNet 53　　(c3) 本文算法

(c) 模糊目标的检测

图 4　本文方法与现有算法对难以检测目标的检测结果比较

高,具有实时性.

后续研究将从两个方面展开:一是采集更多环境下的行人数据集,并以数据增强的方法扩充数据集,使基于该数据集训练后的模型更具泛化能力;二是进一步改进网络结构,以提高模型的检测准确率.

参考文献(References)

[1] 邹逸群, 肖志红, 唐夏菲, 等. Anchor-free的尺度自适应行人检测算法[J]. 控制与决策, 2021, 36(2): 295-302.
(Zou Y Q, Xiao Z H, Tang X F, et al. Anchor-free scale adaptive pedestrian detection algorithm[J]. Control and Decision, 2021, 36(2): 295-302.)

[2] 赵鹏, 徐本朋, 闫石, 等. 基于双分支特征融合的场景文本检测方法[J]. 控制与决策, 2021, 36(9): 2179-2186.
(Zhao P, Xu B P, Yan S, et al. A scene text detection based on dual-path feature fusion[J]. Control and Decision, 2021, 36(9): 2179-2186.)

[3] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. IEEE Conference on Computer Vision and Pattern Recognition. Columbus, 2014: 580-587.

[4] Girshick R. Fast R-CNN[C]. IEEE International Conference on Computer Vision. Santiago, 2015: 1440-1448.

[5] Ren S Q, He K M, 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, 2017, 39(6): 1137-1149.

[6] He K, Gkioxari G, Dollár P, et al. Mask R-CNN[C]. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Venice, 2017: 2961-2969.

[7] Cai Z W, Vasconcelos N. Cascade R-CNN: Delving into high quality object detection[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, 2018: 6154-6162.

[8] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]. IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 779-788.

[9] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]. IEEE Conference on Computer Vision and

Pattern Recognition. Honolulu, 2017: 6517-6525.

[10] Liu W, Anguelov D, Erhan D. SSD: Single shot MultiBox detector[C]. Proceedings of European Conference on Computer Vision. Amsterdam, 2016, 9905: 21-37.

[11] Redmon J, Farhadi A. YOLOv3: An incremental improvement[EB/OL]. (2018-04-08) [2020-02-20]. https://arxiv.org/abs/1804.02767.

[12] 陈泽, 叶学义, 钱丁炜, 等. 基于改进Faster R-CNN的小尺度行人检测[J]. 计算机工程, 2020, 46(9): 226-232.
(Chen Z, Ye X Y, Qian D W, et al. Small-scale pedestrian detection based on improved faster R-CNN[J]. Computer Engineering, 2020, 46(9): 226-232.)

[13] 黄同愿, 杨雪姣, 向国徽, 等. 基于YOLOV3的改进模型在行人检测中的应用[J]. 重庆理工大学学报: 自然科学, 2020, 34(8): 155-164.
(Huang T Y, Yang X J, Xiang G H, et al. Application of improved model based on YOLOV3 in pedestrian detection[J]. Journal of Chongqing University of Technology: Natural Science, 2020, 34(8): 155-164.)

[14] 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.

[15] Ghiasi G, Lin T Y, Le Q V. DropBlock: A regularization method for convolutional networks[J/OL]. 2018, arXiv: 1810.12890.

[16] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 770-778.

[17] He T, Zhang Z, Zhang H, et al. Bag of tricks for image classification with convolutional neural networks[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, 2019: 558-567.

[18] Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[19] 王思元, 王俊杰. 基于改进YOLOv3算法的高密度人群目标实时检测方法研究[J]. 安全与环境工程, 2019, 26(5): 194-200.
(Wang S Y, Wang J J. Dense population real-time detection method based on improved YOLOv3 algorithm[J]. Safety and Environmental Engineering, 2019, 26(5): 194-200.)

[20] Yu J H, Jiang Y N, Wang Z Y, et al. UnitBox: An advanced object detection network[EB/OL]. (2016-08-04) [2020-02-20]. https://arxiv.org/pdf/1608.01471.

[21] Zheng Z H, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[J/OL]. 2019, arXiv: 1911.08287.

[22] Yu F, Xian W, Chen Y, et al. BDD100K: A diverse driving video database with scalable annotation tooling[EB/OL]. (2018-05-12) [2020-02-20]. https://arxiv.org/abs/1805.04687.

[23] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite[C]. IEEE Conference on Computer Vision and Pattern Recognition. Providence, 2012: 3354-3361.

[24] Zhang S S, Benenson R, Schiele B. CityPersons: A diverse dataset for pedestrian detection[C]. IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 4457-4465.

[25] Dong S, Ma Y H, Li C M. Implementation of detection system of grassland degradation indicator grass species based on YOLOv3-SPP algorithm[J]. Journal of Physics: Conference Series, 2021, 1738(1): 012051.

[26] 高星, 刘剑飞, 郝禄国, 等. 基于YOLOv3算法的训练集优化和检测方法的研究[J]. 计算机工程与科学, 2020, 42(1): 103-109.
(Gao X, Liu J F, Hao L G, et al. A training set optimization and detection method based on YOLOv3 algorithm[J]. Computer Engineering & Science, 2020, 42(1): 103-109.)

[27] 陈俊. 基于YOLOv3算法的目标检测研究与实现[D]. 成都: 电子科技大学, 2020.
(Chen J. Research and implementation of target detection based on YOLOv3 algorithm[D]. Chengdu: University of Electronic Science and Technology of China, 2020.)

[28] Bochkovskiy A, Wang C, Liao H M. YOLOv4: Optimal speed and accuracy of object detection[EB/OL]. (2020-4-23) [2020-02-20]. https://arxiv.org/abs/2004.10934.

作者简介

钱惠敏 (1980−), 女, 副教授, 博士, 从事机器学习、目标检测、视频分析与理解等研究, E-mail: qhmin0316@163.com;

陈纬 (1996−), 男, 硕士生, 从事机器视觉、目标检测的研究, E-mail: 191606020014@hhu.edu.cn;

马宜龙 (1998−), 男, 硕士生, 从事机器视觉、目标检测的研究, E-mail: 760448238@qq.com;

施非 (1996−), 男, 硕士生, 从事深度学习、目标检测的研究, E-mail: 1426557795@qq.com;

项文波 (1976−), 男, 讲师, 硕士, 从事图像和视频处理等研究, E-mail: xiang_wb163.com.

(责任编辑：李君玲)