

基于 YOLOv5-MobileNetV3 算法的目标检测^①



曲英伟, 刘 锐

(大连交通大学 软件学院, 大连 116052)

通信作者: 刘 锐, E-mail: 236422217@qq.com

摘 要: 车辆行驶过程中, 对前方目标的检测速度和检测精度一直是自动驾驶领域研究的重点. 针对现有的目标检测算法模型, 在复杂交通环境下, 传统模型面对重叠目标容易导致误检和漏检的情况发生, 大幅度提高检测精度又会造成计算量过大导致处理速度缓慢, 实时性下降的问题. 本文提出基于 YOLOv5 模型的改进算法. 首先采用 MobileNetV3 网络替换原模型中主干网络 C3 的方案, 实现网络仍保持轻量化的同时, 提高模型响应速度. 其次, 提出一种非极大值抑制算法 Adaptive-ElIoU-NMS 来提高重叠目标的识别精度. 最后采用 K-means++ 聚类算法替换原有聚类算法, 生成更精确的锚框. 实验结果表明, 改进后的模型平均检测精度达到 90.1%, 检测速度达到 89 f/s. 实验结果可以证实, 改进后的模型针对复杂场景检测精度和检测速度都有显著提高.

关键词: 自动驾驶; YOLOv5; MobileNetV3; Adaptive-ElIoU-NMS; K-means++

引用格式: 曲英伟, 刘锐. 基于 YOLOv5-MobileNetV3 算法的目标检测. 计算机系统应用, 2024, 33(7): 213-221. <http://www.c-s-a.org.cn/1003-3254/9583.html>

Object Detection Based on YOLOv5-MobileNetV3 Algorithm

QU Ying-Wei, LIU Rui

(School of Software, Dalian Jiaotong University, Dalian 116052, China)

Abstract: The detection speed and accuracy of detecting targets ahead during vehicle operation have always been a focus of research in the field of autonomous driving. For existing object detection algorithm models, in complex traffic environments, traditional models are prone to false positives and missed detections when facing overlapping targets. Significantly improving detection accuracy can also lead to increased computational demands, resulting in slower processing speed and decreased real-time performance. This article proposes an improved algorithm based on the YOLOv5 model. Firstly, the MobileNetV3 network is adopted to replace the C3 backbone network in the original model, achieving a lightweight network while improving the model's response speed. Secondly, a non-maximum suppression algorithm, Adaptive-ElIoU-NMS, is proposed to improve the recognition accuracy of overlapping targets. Finally, the K-means++ clustering algorithm is used to replace the original clustering algorithm and generate more accurate anchor boxes. Experimental results show that the improved model achieves an average detection accuracy of 90.1% and a detection speed of 89 frames per second (f/s). The experimental results confirm that the enhanced model significantly improves both detection accuracy and speed for complex scene detection.

Key words: autonomous driving; YOLOv5; MobileNetV3; Adaptive-ElIoU-NMS; K-means++

^① 收稿时间: 2024-02-05; 修改时间: 2024-03-05; 采用时间: 2024-03-18; csa 在线出版时间: 2024-06-05
CNKI 网络首发时间: 2024-06-07

自动驾驶技术^[1]是当今世界的热门技术,它在汽车工业领域的应用达到了前所未有的程度.随着深度学习^[2]在计算机视觉^[3]领域的深度融合,车辆的智能化不再只是起到辅助驾驶的作用.越来越多的实例表明,在某些场景下,采用自动驾驶技术的车辆可以实现自主安全驾驶.但是,受限于硬件计算能力,深度学习技术需要消耗大量的计算资源.这就要通过改进算法来提升模型的检测速度,同时兼顾检测精度,使自动驾驶技术广泛地运用在车辆或其他交通场景中.

2006年,深度学习的概念被 Hinton 等^[4]首次提出,从而使基于深度学习的计算机视觉技术步入了应用的快车道,目标检测算法迎来高速发展期.基于深度学习算法的目标检测算法分为一阶段 (one stage) 和二阶段 (two stage) 方法,首先二阶段检测算法,代表 R-CNN、Fast R-CNN^[5]、Faster R-CNN^[6]等,这种方法首先需要在遍历区域生成一系列作为样本的候选区域 (region proposal),再通过卷积神经网络 (convolutional neural network, CNN) 来分类.2014年, Girshick 等提出了 R-CNN 检测算法^[7],这种方法训练阶段多,处理速度慢,无法保证图片不变形,总体检测速度慢.2018年, Tian 等^[8]提出改进的 Faster R-CNN 算法,通过采用多层次组合、上下文线索和生成新锚框 (anchor box) 的方法,获得了针对小目标检测时比 R-CNN 更快的检测速度.尽管如此,在实际应用过程中因为计算的复杂性,它们的检测速度仍旧很不理想.其次是一阶段检测算法,代表为 YOLO^[9]、SSD^[10]等.此类算法特点是包含单个神经网络模型,直接在目标图像生成锚框,并进行分类和边界框回归.由于采用单阶段一次完成,因此检测速度比二阶段检测算法有了巨大的提升,但也因为生成的锚框较多,在小目标检测上,容易产生误检、漏检等情况,严重影响检测精度.

目前,在 YOLO 体系下发展而来的算法是应用最广泛的目标检测算法.无数学者对其进行的改造,使其获得更快的速度和更优的精确性.2023年,潘卫华等^[11]通过删减 YOLOv3 网络结构来降低卷积层过多带来的计算复杂问题,改进后计算速度提升效果明显,但在针对小目标检测精度上受到一定影响.2023年,杨志军等^[12]通过裁剪 YOLOv4-tiny 中 CSPResNet 模块一半数量的卷积核,并重新设计空间金字塔池化,极大地提高了检测速度.同时通过改进特征金字塔网络、引入注意

力机制和使用池化特征增强方法,提高整体检测精度.2022年,郭磊等^[13]改进 YOLOv5 算法,通过调整损失函数和修改目标框回归公式,提升了小目标的检测精度.

在改造 YOLO 系列模型过程中,把改造重点放在提高模型检测精度上时,模型中算法或结构会变得复杂,造成计算量过大,从而影响算法的整体检测速度.如果过于追求检测速度的提升,又会造成网络的过度轻量化,虽有效地降低了计算量,但很难兼顾对检测精度的更高要求.

本文依据前人学者的研究成果提出改进 YOLOv5 目标检测算法,主要改进点如下.

(1) 为了提高模型的整体响应速度,采用轻量化网络 MobileNetV3 模型替换 YOLOv5 的主干网络 C3.

(2) 改进原有的非极大值抑制算法 (non-maximum suppression, NMS), 提出 Adaptive-EIoU-NMS 算法,用以返回更精确的目标边界框.

(3) 采用 K-means++ 算法替代 K-means 算法,生成更准确的锚框.

1 相关介绍

1.1 原始 YOLOv5 的结构

作为一阶段目标检测的重要算法, YOLO 系列由 Ultralytics 公司开发, YOLOv5^[14]是 YOLO 系列出现以来最流行的算法模型之一,它的总体性能弱于前代的 YOLOv4^[15],但速度和灵活性上优于 YOLOv4,适合在大多数终端设备上部署. YOLOv5 网络结构^[15]分为 4 部分: 输入层、Backbone (主干网络)、Neck 网络、Head (输出端),如图 1 所示.

输入图片时,采用 Mosaic 数据增强方式,对获取的 4 张图片进行拼接后得到新的图片,提高训练效率.对不同数据集, YOLOv5 可以自适应调节锚框的值,对比 YOLOv4,计算量较少,提升了整体效率.

原始模型采用 C3 网络模型,它采用 3 个标准卷积和 Bottleneck 模块,来加强对残差特征的学习能力. C3 结构有两个形式,如图 2 所示, C3 模块中,内侧循环代表多个 Bottleneck 结构堆叠. C3_F 模块仅经过一个基本卷积模块.合并两种形式进行 Concat 操作.实现了算法轻量化,并保证算法检测精度.

在 Backbone 部分,如图 3 所示,它们对应图 1 中

各重要模块, 通过其内部结构展示, 便于理解其工作原理. 如图 3(a) 所示, 采用 CBS 对信号进行二维卷积, 通过 BN 归一化处理后, 进入激活函数部分. 如图 3(b) 所示 SPP (spatial pyramid pooling) 为空间金字塔池化层,

经过 3 个不同的最大池化层进行下采样, 再拼接融合. 如图 3(c) 所示, 使用 Focus 实现快速下采样, 通过切片操作, 减少计算量, 提高推理速度. 作为主干层, 其作用就是从图像中提取特征信息, 以供后面使用.

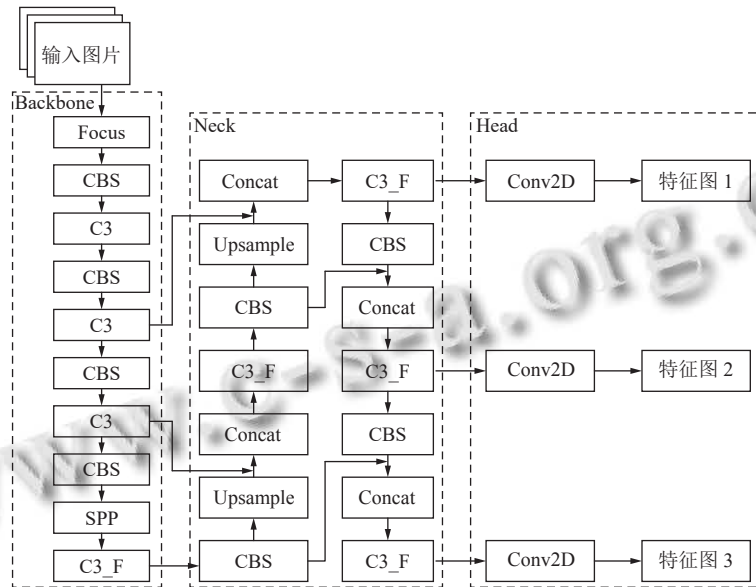


图 1 YOLOv5 的简要结构

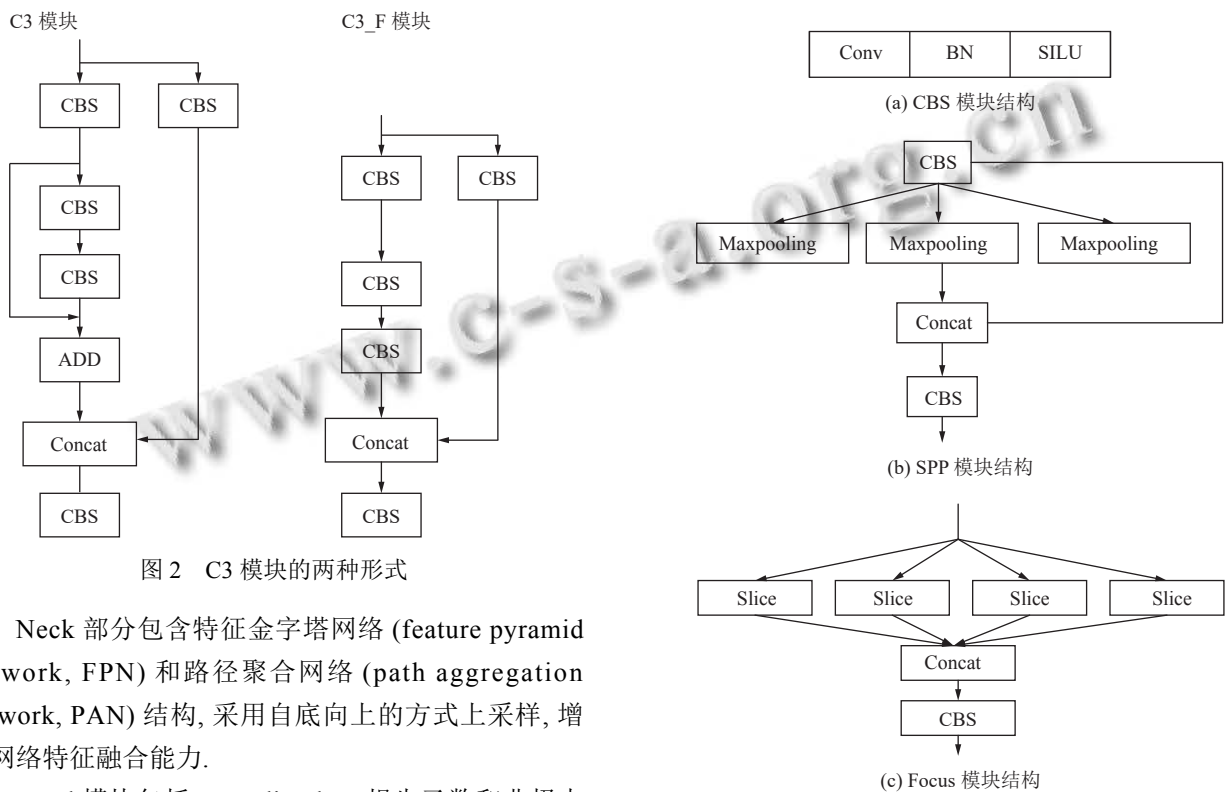


图 2 C3 模块的两种形式

Neck 部分包含特征金字塔网络 (feature pyramid network, FPN) 和路径聚合网络 (path aggregation network, PAN) 结构, 采用自底向上的方式上采样, 增强网络特征融合能力.

Head 模块包括 Bounding box 损失函数和非极大值抑制 (non-maximum suppression, NMS).

图 3 各模块结构

1.2 MobileNetV3 网络简介

MobileNetV3^[16]是由 Google 在 2019 年提出的一种轻量级卷积神经网络结构,因为计算量小,推理时间短的特点,适用于存储和功耗要求受限的场景。

MobileNetV3 结合神经网络架构搜索 (neural architecture search, NAS)^[17]和人工设计的网络结构,采用 Platform-Aware NAS^[18]进行全局搜索得到整体框架,再通过 NetAdapt^[19]搜索进行局部微调的互补策略。

MobileNetV3 有两类核心技术。

(1) 引入 SE 通道注意力模块。它可以保证对输入特征图进行通道特征加强后,不改变输入特征图的大小。常被用在视觉模型中。

SE 注意力模块由两类操作组成,即 S (squeeze) 压缩操作,如图 4(a)。E (excitation) 激励操作,如图 4(b)。其中的 $SERadio$ 是缩放参数,目的是减少通道个数,降低计算量。第 1 个全连接层有 $C \times SERadio$ 个神经元,第 2 个全连接层有 C 个神经元。SE 注意力模块的简单运行过程是:

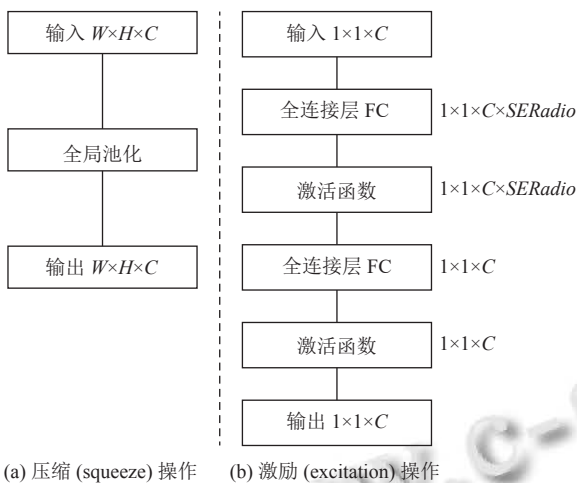


图 4 SE 操作模块

1) 它对输入特征图的空间进行压缩,这是一种全局平均池化操作。

2) 对压缩后的特征图进行特征通道学习,得到具有通道注意力的特征图,此时特征图的大小不变。

3) 将第 2 步得到的特征图和最初输入的特征图,进行逐通道乘以权重系数,最终输出具有通道注意力的特征图。

(2) 采用人工修改的方式,对计算耗时的模块重新设计,提出更快的激活函数 h -swish 如式 (1)。

$$h\text{-swish}[x] = x(\text{ReLU}6(x+3)/6) \quad (1)$$

其中, ReLU 函数是一个取最大值的函数,如式 (2)。

$$\text{ReLU} = \max(0, x) \quad (2)$$

h -swish 函数因为 x 永远大于 0,所以不会存在梯度消失问题。而且他的计算速度很快,只需要判断 $x > 0$ 或 $x < 0$ 这两种情况。收敛速度远远快于其他函数。这也是这种函数出现后被广泛应用的原因。

2 YOLOv5 算法改进

2.1 YOLOv5-MobileNetV3 网络模型

MobileNetV3 有两个版本的网络结构,分别是 large 和 small。MobileNetV3-large 对资源要求相对于 MobileNetV3-small 高,但在准确性和复杂计算上效率高。考虑到检测算法的应用场景复杂,对精度要求高。为了能够兼顾计算的精度和速度,这里最终选择了 MobileNetV3-large 来替换 YOLOv5 网络中的骨干网络 C3 结构,其余部分保持不变。修改后的网络结构如图 5 所示。

经过改进后的 YOLOv5 网络模型结构稳定。MobileNetV3 作为轻量化网络,它的架构也具有扩展性,不必过分受到设备升级的制约,可以通过增加层数和参数量获得更高的准确性。相对于原 YOLOv5 网络,对计算资源的需求非常低。

MobileNetV3 继承前两代的网络结构。即深度可分离卷积对输入端先进行通道分离,再通过逐通道卷积实现。倒残差模块又通过增加网络深度和非线性保持计算代价不变,并且连接两个卷积模块实现了信息跨层传递。这种结构有效地减少参数量和计算量。

2.2 非极大值抑制算法改进

NMS 即非极大值抑制算法^[20],算法通过抑制局部非极大值元素,定位搜索范围内的极大值。这种方法避免了针对同一个目标的多次检验,同时也可以确保每个目标获得一次检测。如图 6 所示,在 NMS 处理之前,会先通过置信度阈值清理过多重复的候选框 (candidate bound),再通过 NMS 算法选取置信度最大的边界框定位检测目标^[21]。

传统 NMS 中,预测框和真实框算出对应的交并比 (intersection over union, IoU) 值,在这里 IoU 作为唯一的评价指标,面对多个检测对象重叠, IoU 值较大时,经过处理后只剩一个检测框,从而导致漏检情况。为保证检测精度本文提出新算法 Adaptive-ElIoU-NMS,其

中使用 $EIoU^{[22]}$ 作为距离的评判指标, $EIoU$ 损失函数将预测框和真实框的纵横比影响因子拆开分别计算长

度和宽度, 使预测框和真实框的长宽之差最小, 提高了预测框收敛速度和回归精度。

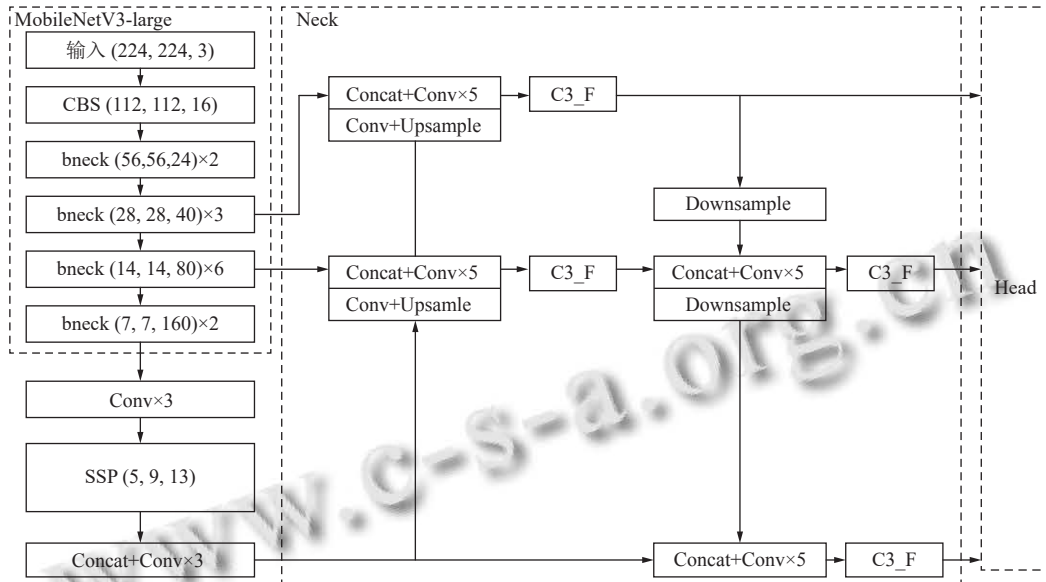


图5 YOLOv5-MobileNetV3-large 结构



图6 锚框定位

$EIoU$ 损失函数公式如下:

$$LEIoU = L_{IoU} + L_{dis} + L_{asp}$$

$$= 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{C_w^2} + \frac{\rho^2(h, h^{gt})}{C_h^2} \quad (3)$$

其中, 预测框和真实框交并损失 L_{IoU} 、中心距离损失 L_{dis} 和宽高损失 L_{asp} 之和组成 $EIoU$. b 是真实框中心点坐标, b^{gt} 是预测框中心点坐标, $\rho(b, b^{gt})$ 表示预测框和真实框中心点之间的欧氏距离, C_w 和 C_h 是预测框和真实框最小矩形宽高, c^2 表示 $C_w^2 + C_h^2$. w 是真实框的宽, w^{gt} 是预测框的宽, h 是真实框的高, h^{gt} 是预测框的高. $\rho(h, h^{gt})$ 和 $\rho(w, w^{gt})$ 分别表示预测框和真实框的高之间的欧氏距离和宽之间的欧氏距离。

Adaptive-NMS^[23] 是对传统 NMS 的改进, 是一种自适应调节抑制极大值阈值的方法. 这里添加了密度预测模块, 这样在检测严重重叠目标时, 判断重叠物体的密度程度, 这时 Adaptive-NMS 密度定义如下:

$$d_i = \max_{b_j \in \mathcal{C}, i \neq j} IoU(b_i, b_j) \quad (4)$$

其中, i 表示所有预测框, j 表示所有真实框, d_i 表示第 i 处的目标密度, 通过计算出全部预测框 b_i 和全部真实框 b_j 的最大 IoU 来表示密度。

阈值定义如下:

$$N_M = \max(N_t, d_M) \quad (5)$$

其中, N_M 表示置信度得分最高预测框 M 的阈值, N_t 为人为设定的剪除多余框时阈值, d_M 是 M 的目标密度, 通过求它们的最大值得到结果。

把 $EIoU$ 同 Adaptive-NMS 结合, 置信度惩罚公式如下:

$$f(EIoU(M, b_i)) = e^{-\frac{EIoU(M, b_i)^2}{\sigma}} \quad (6)$$

$$S_i = \begin{cases} s_i, & EIoU(M, b_i) < N_M \\ s_i f(EIoU(M, b_i)), & EIoU(M, b_i) \geq N_M \end{cases} \quad (7)$$

式(6)和式(7)中各变量在前述中各有表述, 这里不再赘述, 在这里将结合后的抑制算法分3种情形分析。

(1) M 周围不存在预测框时($EIoU < N_t$), s_i 不变。

(2) M 周围存在密集预测框时($N_M > N_t$), d_M 成为抑制阈值。

(3) M 周围存在稀疏预测框时($N_M \leq N_t$), N_t 成为抑制阈值。

2.3 K-means++替换原网络聚类算法

YOLOv5采用锚框来预测待检测目标, 锚框的尺寸依据检测目标的长宽比得到。YOLOv5网络的默认锚框是依据COCO数据集训练过程中得到的。根据K-means^[24]聚类算法特点, 聚类前需要初始化 K 个簇核心。这就意味着YOLOv5的聚类收敛情形依赖于COCO数据集的簇核心的初始化状态。因为YOLOv5模型的锚框差距较大, 为了更好地对小目标进行检测, 根据新的数据集, 需要进一步的调整重新聚类。

另外, K-means算法中 K 的取值对于聚类结果和运行时间影响巨大。如果完全随机选择, 有可能导致算法收敛缓慢, 时间增加, 计算量增大, 算法精度下降等问题。

所以, 在这里引入K-means++算法^[25]来替换原网络中的K-means算法。以此算法来重新对数据集中的边界框聚类, 从而得到合适的锚框。

K-means++算法优化了K-means算法中 K 个聚类中心的初始化方法。大大分散了初始的聚类中心之间的距离。

基本算法如下:

(1) 在数据集 $X = \{x_1, \dots, x_n\}$ 中选择一个随机数据作为初始聚类中心 a_1 。

(2) 计算每个样本点 x_i 与所有聚类中心的最短路径。用 $D(x_i)$ 表示。计算每个样本点被选为新的聚类中心的概率 $D(x)^2 / \sum_{x \in X} D(x)^2$ 。

(3) 重复步骤(2)直至选择出共 K 个聚类中心 $A = \{a_1, \dots, a_k\}$ 。

(4) 针对每个样本 x_i , 计算 x_i 到 K 个聚类中心的距离, 并将 x_i 分到相距最近聚类中心的类别中。

(5) 针对聚类中心所在类别 a_i , 重新计算它的聚类中心 $a_1 = \left(\sum_{x \in a_i} x \right) / |a_i|$ 。

(6) 重复步骤(5)和步骤(6)直至聚类中心稳定。

3 实验与分析

3.1 实验环境

本实验在Ubuntu 20.04系统中运行, CPU采用Intel酷睿i7-12700H, 内存32GB, GPU采用NVIDIA GeForce RTX3080Ti, 显存12GB, 加速环境为CUDA 11.2.0, CUDNN 8.1.1, 集成开发环境为PyCharm专业版2023.2, 编程语言Python 3.8。

3.2 实验数据集和设置

本次实验采用KITTI数据集, 这是一款在自动驾驶领域应用最广泛的经典数据集。数据集中包含大量的目标重叠场景和小目标。原始数据集中细分为9类标签, 考虑到实际情况, 将Car、Van、Truck、Tram统一归类为Car, 将Pedestrian、Person_setting统一归类为Pedestrian, 保留Cyclist和Misc, 删除未标注的DontCare。本实验数据集采纳7481张已标注图片, 按照8:1:1的比例将数据集分为训练集、测试集、验证集, 得到训练集图片5985张, 测试集图片748张和验证集图片748张。

本实验采用深度学习框架TensorFlow-GPU 2.7.0, Keras 2.7.0对改进前后的模型进行训练。输入图片大小按照YOLOv5的默认值设定为640×640, 批处理大小batchsize为16, 初始学习率设定为0.01, 权重衰减系数为0.0002, IoU 阈值为0.5, epoch为200。

3.3 评价指标

我们将在实验的不同阶段引入不同的评价指标作为参考, 进行多维度分析。在消融实验阶段, 采用 $mAP_{0.5}$ 代表 IoU 阈值为0.5时的平均 AP (average precision) 和 $mAP_{0.5:0.95}$ 表示 IoU 阈值为0.5–0.95, 步长

为 0.05 的 mAP (mean average precision) 平均值, 我们也引入常用评估指标精确度 P (precision) 和召回率 R (recall). 在对比试验阶段, 加入 mAP 和帧率 FPS (frame per second) 的值作为最终算法性能的量化指标.

计算公式如下所示:

$$P = \frac{TP}{TP + FP} \quad (8)$$

$$R = \frac{TP}{TP + FN} \quad (9)$$

$$AP = \int_0^1 P(r)dr \quad (10)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (11)$$

式 (8)–式 (11) 中, TP 表示 ($IoU \geq$ 某阈值数) 的预测框数, FP 表示 ($IoU \leq$ 某阈值数) 的预测框数, FN 表示漏检的目标数.

3.4 消融实验

本文提出了针对传统 YOLOv5 的一种改进方法. 改进的核心是更换主干网络 MobileNetV3、改进非抑制最大值算法为 Adaptive-Elou-NMS, 替换原模型中聚类算法 K-means 为 K-means++. 为了研究各模块带来的具体何种性能改变, 需要逐个验证效果. 这里以 YOLOv5 原始算法为基础, 按照不同的组合顺序删改相关模块, 根据实验数据, 判断改进后的算法是否能够达到预期. 本文在 KITTI 数据集上进行消融实验, 实验结果如表 1.

表 1 算法消融实验结果 (%)

| 组别 | Mobile NetV3 | Adaptive-Elou-NMS | K-means ++ | $mAP_{0.5}$ | $mAP_{0.5:0.95}$ | P | R |
|----|--------------|-------------------|------------|-------------|------------------|------|------|
| 1 | × | × | × | 88.5 | 63.1 | 89.1 | 86.5 |
| 2 | √ | × | × | 88.9 | 63.9 | 89.5 | 86.9 |
| 3 | × | √ | × | 90.0 | 65.3 | 90.8 | 87.6 |
| 4 | × | × | √ | 88.6 | 63.4 | 89.2 | 87.1 |
| 5 | √ | × | √ | 88.5 | 64.8 | 90.2 | 86.6 |
| 6 | × | √ | √ | 89.6 | 65.5 | 90.9 | 88.0 |
| 7 | √ | √ | × | 90.3 | 66.1 | 91.0 | 87.5 |
| 8 | √ | √ | √ | 90.4 | 66.8 | 91.3 | 88.4 |

依据实验数据, 由第 2 组实验我们可以看到, 主干网络完成了轻量化替换, 对应的 $mAP_{0.5}$ 和 $mAP_{0.5:0.95}$ 分别提高了 0.4% 和 0.8%, 这是由于 MobileNetV3 网络模型增加了层数和参数量, 整体的计算量有所增加, 进而提高计算的精确性. 再看第 3 组, $mAP_{0.5}$ 提高了

1.5%、 $mAP_{0.5:0.95}$ 提高了 2.1%, 这是本文的重点修改部分. 对 NMS 进行了改造, 把 $Elou$ 的引入进算法, 有效地解决了目标重叠时, 因为无法区分而造成的漏判, 同时新增加了自适应选择阈值, 提高了算法的灵活性. 同时观察第 4 组, 在这里引入 K-means++ 算法, 我们可以看到, 相较于前两组的提升, 这里改造后的算法 $mAP_{0.5}$ 提升了 0.1%, $mAP_{0.5:0.95}$ 提升了 0.3%, 我们可以看到, 虽然新算法相较于传统算法有了很大提升, 但是实验表现并不突出.

同时, 我们在消融实验表中也可以看到, 虽然各个模块都有优势的地方, 但组合在一块的时候, 并不能发挥很明显的效果. 例如第 5 组数据, 虽然更新了两个模块, 但并没有达到预期, 不如第 3 组提升的明显, 甚至还有倒退. 但总体上来说, 改造后的新算法提升效果明显, $mAP_{0.5}$ 提高了 2.9%, $mAP_{0.5:0.95}$ 提高了 3.7%. 实现了面对复杂道路情况的精准检测.

3.5 对比实验和综合性能分析

为了进一步验证新模型与其他模型比较后性能提升效果, 采用数据集 KITTI 对不同模型进行对比测试, 本文将采用近年主流目标检测算法模型, 采用同样的参数标准进行训练、测试和验证. 并进行全面的分析对比, 验证改进算法的合理性和有效性.

从表 2 可知, 二阶段算法 Faster R-CNN 在同数据集同参数设定情况下, 检测精度明显优于传统的一阶段算法, 如表 2 中的 SSD、YOLOv3、YOLOv4、YOLOv5 等. 二阶段算法具有两个训练步骤, 漏检错检率低, 精度高, 但计算复杂, 实时性差. 可以看到, 二阶段算法的帧率相较于一阶段算法差距明显. 另一方面, 一阶段算法中两个主流算法是 SSD 和 YOLO 家族, 在这里, 因为 YOLO 家族引入了 FPN 结构, 检测层由三级要素图层 (feature layers) 融合, 并在浅层包含大量小目标特征, 无论是整体检测精度还是针对小目标的检测精度, 都明显优于 SSD 的仅有一级要素图层的检测精度. 这点从表 2 中关于平均精度的数值变化有明显的体现. 每一代 YOLO 版本都各有优缺点, 理论上, 精度高的算法模型因为计算量过大, 很容易导致检测速度变慢. YOLO 在这里运用轻量化主干网络模型来提高检测速度, 同时不断的改进内部网络结构来提升检测精度. 在 YOLO 家族的较早期版本中, 采用不同的主干网络, 对整体模型效果有着巨大的影响, 比如 YOLOv3 和 YOLOv4, 在 YOLOv3 中采用 Darknet53 这种全卷

积结构,对比 YOLOv4 采用的 CSPDarknet 结构因为缺少跨层信息连接,在计算复杂度上大大增加,这就使得 YOLOv3 检测精度和速度都相对之后版本较弱.即使在同版本的情况下,主干网络的不同,模型的性能差异也是很明显的.比如在 YOLOv4 和它的改进型 YOLOv4-tiny 上,后者是前者的简化版本,因为 YOLOv4-tiny 在特征提取时删除了 Mish 激活函数,并仅采用一个特征金字塔,所以在计算量上大大简化,代价是精度有所下降.我们在实验中看到, YOLOv4-tiny 平均检测精度比 YOLOv4 低了 4.4%,但帧率却提高了 58 f/s,这表明检测精度和检测速度无法有效兼顾,但在可行范围内可以做到平衡.同理,我们也可以在 YOLOv5 和 YOLOv5m 上看到这种问题.

表 2 对比实验结果

| 算法 | 主干网络 | 输入尺寸 | mAP (%) | FPS |
|----------------------------|--------------|----------|---------|-----|
| Faster R-CNN | ResNet101 | 1000×600 | 87.5 | 14 |
| SSD | VGG16 | 512×512 | 73.2 | 49 |
| YOLOv3 ^[26] | Darknet53 | 416×416 | 79.2 | 57 |
| YOLOv4 ^[27] | CSPDarknet53 | 416×416 | 86.5 | 74 |
| YOLOv4-tiny | CSPDarknet53 | 416×416 | 82.1 | 91 |
| YOLOX-tiny ^[28] | Darknet53 | 640×640 | 88.1 | 52 |
| YOLOv5 | CSPDarknet53 | 640×640 | 85.3 | 95 |
| YOLOv5m | CSPDarknet53 | 640×640 | 88.9 | 63 |
| Ours | MobileNetV3 | 640×640 | 90.1 | 89 |

依据前人的经验和算法迭代特点,本文对主干网络的改进,使得检测精度和检测速度不做过多的妥协.本文算法模型采用的 MobileNetV3 网络作为主干网络,克服了传统网络的缺点,创新采用的互补搜索技术,这种技术需要的资源少,保证了在低配置终端上的使用,又对参数量要求少,同时在模块间进行搜索,大大降低了计算复杂度,提高了速度.进一步重新设计了网络结构,减少卷积层使得网络结构整体更趋于轻量化,改进的 swish 激活函数使得计算复杂度进一步降低,模型整体计算速度更快,但考虑到这种修改往往会牺牲检测精度,引入的 Adaptive-NMS 和 EIou 结合后的新算法,一边采用自适应的抑制策略,通过判断目标周边密度信息,以达到对重叠目标和小目标的精准定位.最后采用更优化的 K-means++ 算法,生成更加高效的锚框,避免了随机选择导致的聚类中心过近,大量接近的重复计算过多,增加整体网络负担.通过这些改变,在表 2 中体现明显,检测精度和检测速度达到了平衡,相比对原模型 YOLOv5,改进后算法检测精度提高了 4.7%,检测速度略有下降,帧率仅下降 6 f/s.通过图 7 展示效果和实验数值可以看出,相较于其他算法,无论是检测精度还是检测速度都提升明显,说明本文的改进模型性能效果良好.



图 7 检测效果图

4 结束语

复杂的场景会对相关传感器造成严重的干扰,图像中存在大量的噪声和遮蔽信息.本文采用 MobileNetV3 网络作为骨干网络提高检测速度、提出 Adaptive-EIoU-NMS 来提高检测精度和采用 K-means++ 聚类算法生成新锚框等操作,得到新的 YOLOv5 网络模型.实验表明了这一方法的有效性.但在强光照和大雾雨雪天气下,对于小目标仍存在错检和漏检的情况.未来研究工作的重点就是进一步提高模型在更加复杂情况下的检测准确性和实时性上.

参考文献

- 1 李川鹏,郭宇辰.浅析自动驾驶技术发展现状、趋势及挑战.时代汽车,2022(14):4-6.
- 2 李柯泉,陈燕,刘佳晨,等.基于深度学习的目标检测算法综述.计算机工程,2022,48(7):1-12. [doi: 10.19678/j.issn.1000-3428.0062725]
- 3 肖久源,吴芳榕.自动驾驶汽车中视觉技术的应用.汽车测试报告,2022(19):43-45.
- 4 Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science, 2006, 313(5786): 504-507. [doi: 10.1126/science.1127647]

- 5 Girshick R. Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). Santiago: IEEE, 2017. 1440–1448. [doi: [10.1109/iccv.2015.169](https://doi.org/10.1109/iccv.2015.169)]
- 6 Ren SQ, He KM, Girshick R, *et al.* Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149. [doi: [10.1109/TPAMI.2016.2577031](https://doi.org/10.1109/TPAMI.2016.2577031)]
- 7 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. [doi: [10.1109/CVPR.2014.81](https://doi.org/10.1109/CVPR.2014.81)]
- 8 Tian Y, Gelernter J, Wang X, *et al.* Lane marking detection via deep convolutional neural network. Neurocomputing, 2018, 280: 46–55. [doi: [10.1016/j.neucom.2017.09.098](https://doi.org/10.1016/j.neucom.2017.09.098)]
- 9 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. [doi: [10.1109/CVPR.2016.91](https://doi.org/10.1109/CVPR.2016.91)]
- 10 Liu 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. [doi: [10.1007/978-3-319-46448-0_2](https://doi.org/10.1007/978-3-319-46448-0_2)]
- 11 潘卫华, 罗裕坤. 基于改进 YOLOv3 的车辆目标检测. 计算机应用与软件, 2023, 40(1): 167–172, 204. [doi: [10.3969/j.issn.1000-386x.2023.01.027](https://doi.org/10.3969/j.issn.1000-386x.2023.01.027)]
- 12 杨志军, 昌新萌, 丁洪伟. 基于改进 YOLOv4-Tiny 的交通车辆实时目标检测. 无线电工程, 2023, 53(11): 2635–2644. [doi: [10.3969/j.issn.1003-3106.2023.11.019](https://doi.org/10.3969/j.issn.1003-3106.2023.11.019)]
- 13 郭磊, 王邱龙, 薛伟, 等. 基于改进 YOLOv5 的小目标检测算法. 电子科技大学学报, 2022, 51(2): 251–258.
- 14 邵延华, 张铎, 楚红雨, 等. 基于深度学习的 YOLO 目标检测综述. 电子与信息学报, 2022, 44(10): 3697–3708. [doi: [10.11999/JEIT210790](https://doi.org/10.11999/JEIT210790)]
- 15 王鹏飞, 黄汉明, 王梦琪. 改进 YOLOv5 的复杂道路目标检测算法. 计算机工程与应用, 2022, 58(17): 81–92. [doi: [10.3778/j.issn.1002-8331.2205-0158](https://doi.org/10.3778/j.issn.1002-8331.2205-0158)]
- 16 Howard A, Sandler M, Chen B, *et al.* Searching for MobileNetV3. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul: IEEE, 2020. 1314–1324. [doi: [10.1109/ICCV.2019.00140](https://doi.org/10.1109/ICCV.2019.00140)]
- 17 Elsken T, Metzen JH, Hutter F. Neural architecture search: A survey. The Journal of Machine Learning Research, 2019, 20(1): 1997–2017.
- 18 Tan MX, Chen B, Pang RM, *et al.* MnasNet: Platform-aware neural architecture search for mobile. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach: IEEE, 2018. 2815–2823. [doi: [10.1109/CVPR.2019.00293](https://doi.org/10.1109/CVPR.2019.00293)]
- 19 Yang TJ, Howard A, Chen B, *et al.* NetAdapt: Platform-aware neural network adaptation for mobile applications. Proceedings of the 15th European Conference on Computer Vision. Munich: Springer, 2018. 289–304. [doi: [10.1007/978-3-030-01249-6_18](https://doi.org/10.1007/978-3-030-01249-6_18)]
- 20 Neubeck A, Van Gool LJ. Efficient non-maximum suppression. Proceedings of the 18th International Conference on Pattern Recognition. Hong Kong: IEEE, 2006. 850–855. [doi: [10.1109/ICPR.2006.479](https://doi.org/10.1109/ICPR.2006.479)]
- 21 李建新, 陈厚权, 范文龙. 基于改进 YOLOv5 的遥感图像目标检测研究. 计算机测量与控制, 2023, 31(9): 102–108, 115.
- 22 杨睿宁, 惠飞, 金鑫, 等. 改进 YOLOv5s 的复杂交通场景路侧目标检测算法. 计算机工程与应用, 2023, 59(16): 159–169. [doi: [10.3778/j.issn.1002-8331.2304-0090](https://doi.org/10.3778/j.issn.1002-8331.2304-0090)]
- 23 Liu ST, Huang D, Wang YH. Adaptive NMS: Refining pedestrian detection in a crowd. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach: IEEE, 2019. 6452–6461. [doi: [10.1109/CVPR.2019.00662](https://doi.org/10.1109/CVPR.2019.00662)]
- 24 杨昊瑜, 戴华林, 王丽, 等. 基于 K-means 的 ORB 算法在前方车辆识别中的应用. 传感器与微系统, 2020, 39(10): 157–160. [doi: [10.13873/J.1000-9787\(2020\)10-0157-04](https://doi.org/10.13873/J.1000-9787(2020)10-0157-04)]
- 25 吴丽娟, 黄尧, 关贵明. 基于改进 YOLOv5 的道路目标检测. 沈阳师范大学学报(自然科学版), 2023, 41(1): 85–91.
- 26 Liu BK, He FJ, Du SQ, *et al.* An advanced YOLOv3 method for small object detection. Journal of Intelligent and Fuzzy Systems, 2023, 45(4): 5807–5819. [doi: [10.3233/JIFS-224530](https://doi.org/10.3233/JIFS-224530)]
- 27 Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934, 2020.
- 28 郎磊, 刘宽, 王东. 基于 YOLOX-Tiny 的轻量级遥感图像目标检测模型. 激光与光电子学进展, 2023, 60(2): 353–363. [doi: [10.3788/LOP212699](https://doi.org/10.3788/LOP212699)]

(校对责编: 张重毅)