计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(4):149−160 [doi: 10.15888/j.cnki.csa.009048]

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

基于 GhostNet 与注意力机制的 YOLOv5 交通目标检测①

皇甫俊逸, 孟　乔, 孟令辰, 谢宇鹏

(青海大学 计算机技术与应用系, 西宁 810016)
通信作者: 孟　乔, E-mail: 250345481@qq.com

摘　要: 针对交通目标检测模型参数量大、检测精度低、检测速度慢、泛化性差等问题, 提出一种基于 GhostNet 与注意力机制的 YOLOv5 交通目标实时检测模型. 采用基于遗传算法的 K-means 聚类方法获取适用于车辆检测的最佳预选框; 采用轻量的 Ghost 卷积提取目标特征, 并构建基于 CSP 结构的 C3Ghost 模块, 大幅度压缩模型参数量, 降低计算成本, 提高计算速度; 在特征融合层添加 Transformer block 和 CBAM 注意力模块, 来探索模型特征提取潜力以及为模型在密集对象的场景中寻找注意力区域; UA-DETRAC 数据集上的消融实验和综合性能评价结果表明所提模型平均精度达到 98.68%, 参数量为 47 M, 检测速度为 65 FPS, 与 YOLOv5 相比, 参数量压缩了 34%, 速度提升 43%, 平均精度提高了 1.05%.

关键词: 目标检测; 注意力机制; 轻量化网络; YOLOv5; 跨阶段局部网络

引用格式: 皇甫俊逸,孟乔,孟令辰,谢宇鹏.基于 GhostNet 与注意力机制的 YOLOv5 交通目标检测.计算机系统应用,2023,32(4):149–160. http://www.c-s-a.org.cn/1003-3254/9048.html

YOLOv5 Traffic Object Detection Based on GhostNet and Attention Mechanism

HUANGFU Jun-Yi, MENG Qiao, MENG Ling-Chen, XIE Yu-Peng

(Department of Computer Technology and Applications, Qinghai University, Xining 810016, China)

Abstract: Traffic object detection models have massive parameters, low detection accuracy and speed, and poor generalization. In view of these problems, YOLOv5 real-time traffic object detection model based on GhostNet and attention mechanism is proposed. The K-means clustering method based on genetic algorithms is used to obtain the best prior bounding box suitable for vehicle detection. The lightweight GhostConv is used to extract target features, and the C3Ghost module based on the CSP structure is constructed, which can greatly reduce the number of model parameters, reduce the calculation cost, and improve the calculation speed. Transformer block and CBAM attention module are added in the feature fusion layer to explore the potential of feature extraction of the model and find attention regions for the model in scenarios with dense objects. The results of ablation experiments and comprehensive performance evaluation on the UA-DETRAC data set show that the average accuracy of the proposed model reaches 98.68%, the number of parameters is 47 M, and the detection speed is 65 FPS. Compared with YOLOv5, the number of parameters is reduced by 34%, the speed is increased by 43%, and the average accuracy is increased by 1.05%.

Key words: object detection; attention mechanism; lightweight network; YOLOv5; cross stage partial network (CSPNet)

① 基金项目: 青海大学中青年基金 (2019-QGY-15)
收稿时间: 2022-09-20; 修改时间: 2022-10-19; 采用时间: 2022-11-04; csa 在线出版时间: 2023-02-24
CNKI 网络首发时间: 2023-02-26

近年来, 计算机视觉技术的快速发展, 广泛应用于自动驾驶, 无人机检测以及安全检测等领域[1]. 随着科技发展、生活水平提高, 人们对私家车、公交车以及卡车需求的增加导致交通拥堵、交通事故等问题愈加严重, 给城市交通监管带来巨大压力. 道路视频监控系统的广泛应用以及面向视频的目标检测技术的不断发展, 使智能化的交通管理成为可能, 而交通目标检测则是智能交通系统中至关重要的研究内容之一, 同时, 也是保证实时获取交通监控信息及构建智能交通系统的先决条件.

传统的车辆检测方法主要分为基于图像和基于视频两类方法. 基于视频的方法包括背景差分法、光流法和帧间差分法, 背景差分法基本思想是将输入图像与背景模型比较, 分割出背景图像和运动目标, 经典算法有 MOG2 (mixture of Gaussians)[2] 和 GMG (geometric multi gid)[3], 该算法实现简单, 且实时性好, 但模型鲁棒性差, 动态环境变化对结果影响较大. 光流法通过一段时间内像素的变化, 实现对运动对象的检测, 常用的有稠密光流[4]、稀疏光流[5], 光流法无须了解场景信息, 就可以准确检测识别运动目标, 但计算量大且抗噪声能力差. 帧间差分法将相邻两帧做差分运算得出运动目标轮廓, 模型的实时性好, 能适应动态变化的场景, 但对于运动慢的目标, 前后帧重合, 从而导致提取目标不完整. 基于图像的方法通常分为 3 部分: 区域选择、特征提取和分类器, 采用滑动窗口的策略遍历整幅图像, 在选择的区域中提取车辆特征, 常用的特征有形状、颜色、对称、纹理、车辆阴影、车辆的前灯和尾灯、边缘等, 使用 HOG[6]、SIFT[7]、Haar[8] 等算子提取特征, 一旦特征被提取出来, 就可以使用诸如 SVM[9]、AdaBoost[10] 等分类器算法进行车辆分类. 然而, 该类方法中但特征提取通常具有随机性和单一性, 且容易受到反射、弱光、天气条件和物体运动等因素的影响. 因此, 传统检测方法设计困难、检测效率低、检测准确性也不高, 在简单交通场景下的车辆检测效果尚可, 但其性能远远无法满足在现实情况下复杂交通环境的车辆检测.

为了克服复杂交通环境带来的挑战, 研究人员将深度学习引入车辆检测领域. 基于深度学习的检测算法大致分为两类: 一种是基于区域建议的目标检测算法, 也称作二阶段检测算法, 如 R-CNN[11], SPP-Net[12], Faster R-CNN[13] 等. 黄继鹏等人[14] 通过改进 Faster-RCNN 网络结构, 实现对小目标的多尺度检测. 实验结果证明了该算法在小目标检测任务上拥有更高的平均精度均值. 陈飞等人[15] 改进 Faster-RCNN 检测模型应用于道路目标检测, 并通过实验证明了该方法相比 Faster-RCNN 具有更高的目标检测准确率. 另一种是基于回归的目标检测算法, 也称作单阶段检测算法, 如 YOLO[16], SSD[17] 等. 该类算法是将目标检测看作回归问题, 如 YOLO 算法是直接对图片中预设的锚框进行类别的预测和回归, 拥有非常快的检测速度, 但检测精度相对较低. 孟乔[18] 设计了一种多层卷积特征融合与金字塔式多尺度计算的 SSD 车辆检测方法. 通过实验证明了所提出方法在精度与速度等多种评估指标下的综合检测性能极佳. 张新宇等人[19] 在 YOLOv4 模型中中引入注意力机制, 并验证了该方法相比于 Faster R-CNN 和 YOLO 系列算法在检测精度和速度上都具有明显优势. 基于区域建议的目标检测算法拥有较高的检测精度, 但检测速度慢, 训练时间长, 内存占用多. 基于回归的目标检测算法速度快, 但检测精度不够. 交通目标检测需要实时性和准确性兼顾, 当目标处于密集、低光照等复杂的现实环境时, 现有交通目标检测算法的精度、速度与实际需求仍有差距. 首先, 交通图像中背景复杂, 交通目标尺寸变化大、重叠遮挡严重, 运动物体成像容易发生形变等因素, 导致交通目标检查精度低; 其次, 用于交通监控的目标检测要满足的实时性需求, 对出现的交通事故、交通违规行为及时识别处理; 同时, 交通图像存在图片分辨率高、尺寸大的情况, 使用传统的卷积计算会导致提取的特征信息冗余, 计算量大, 训练慢, 模型参数大的问题. 综合以上问题, 交通目标检测算法需要平衡检查精度、速度以及模型参数 3 方面性能, 现有的算法很难满足需求.

为应对复杂的交通环境, 本文提出基于 GhostNet[20] 与注意力机制的 YOLOv5[21] 车辆检测模型, 从检测速度、精度、模型大小、泛化性能等角度, 多方面优化 YOLOv5 模型. 本文主要创新及贡献如下.

(1) 设计轻量化网络, 利用 GhostNet 思想消除特征图冗余, 与 CSP[22] 结构融合, 构造出 C3Ghost 模块, 有效减少模型参数量, 降低模型所占资源和计算成本.

(2) 添加注意力机制, 抵消使用 C3Ghost 带来的特征图通道间相关性不足, 加入 Transformer block[23] 和 CBAM[24] 注意力模块, 前者帮助模型获取全局信息, 后者使模型可以在密集对象的场景中找到注意力区域.

(3) 在 UA-DETRAC[25] 公共数据集上, 设计消融对比实验, 分析各模块对 YOLOv5 性能影响, 并使用基于

遗传算法的 K-means 聚类[26]、DropBlock[27] 正则化等技术, 经过不断调参获取到最终模型. 实验结果表明, 对比 YOLOv5 算法, 多方面性能得到提升.

1 基于 GhostNet 与注意力机制的 YOLOv5 交通目标检测方法

1.1 YOLOv5 算法结构

YOLO 算法是目标检测领域第 1 个单阶段检测器, 抛弃之前先提取物体区域再分类识别的检测范式, 提出一种完全不同的理念: 用一个神经网络将图片划分为多个区域, 并同时预测每个区域的边界框和置信度. 尽管 YOLO 拥有很高的检测速度, 但与二阶段检测器相比, 准确度和召回率有所下降. 之后, 人们在 YOLO 的基础上进行一系列的改进, 将优秀的目标检测技术融入 YOLO 算法中, 在保证较高检测速度的同时, 提高了检测的精度. YOLOv5 是 YOLO 系列新一代模型, 添加了一些巧妙的改进思路, 进一步提升检测的速度和精度. YOLOv5 官方给出在 MS COCO (Microsoft common objects in context) 上各模型的性能如图所示, 最快的模型在 CPU 上检测速度达到 22 FPS, 在 V100 GPU 上检测速度达到了 158 FPS, 获得了 45.7% *mAP*, 精度最高的模型以 82 FPS 的速度获得 68.9% *mAP*. 基于以上 YOLOv5 的优越性, 本文选择 YOLOv5 检测网络作为研究对象.

YOLOv5 整体可以分成 4 个模块, 如图 1 所示分别是输入端、Backbone 网络、Neck 网络与 Head 输出端. 输入端主要完成图像预处理工作, 即缩放图片为网络输入大小、数据归一化以及一些数据增强技术. YOLOv5 使用 Mosaic 数据增强方式, 通过将 4 张不同的图片随机剪裁、排布、拼接成一张新的图片. 同时, 作者改进了自适应图片缩放方式, 以往 YOLO 将不规则图片固定地缩放填充为规则的正方形, 存在填充过多, 造成信息冗余, 影响训练、推理的速度, 因此修改缩放方法, 采用自适应的填充方式, 对原始图片添加最少的黑边. Backbone 网络负责提取通用的特征表示, 在 YOLOv5 中作者借鉴了 CSPNet 的设计思想, 先将基础层的特征映射一分为二, 然后通过跨阶段层次结构合, 设计了新型的 CSPDarknet53 结构, 减少了推理中的计算量, 降低了模型的内存成本和计算瓶颈. 值得一提的是, 在初始的 YOLOv5 版本中网络第 1 层使用 Focus 结构进行切片操作, 而在之后换成了 6×6 大小的卷积层, 并说明了两者理论上等价, 且在 GPU 上卷积比 Focus 结构更加高效. Neck 网络对提取的特征加工, 提升特征的多样性, 首先对 SPP 做出改进, 将并行通过不同大小的最大池化层改为串行通过 3 个 5×5 大小的最大池化层, 原理是串行两个 5×5 的最大池化层等于一个 9×9 的最大池化层, 串行 3 个 5×5 的最大池化层输出结果与一个 13×13 的最大池化层一致, 并由实验得出修改的 SPP 结构速度为原来的 2 倍. 其次, 将 FPN (feature pyramid network)[28] 和 PAN (path aggregation network)[29] 结合起来, FPN 自顶向下把深层特征的语义信息传递到浅层, PAN 自底向上将浅层的定位信息传递给深层, 在多尺度上进行预测. Head 检测层大体上延用了之前的 YOLOv3 检测头, 分类损失和置信度损失采用 *BCE loss*, 定位损失采用 *CIoU loss*, 并且为了消除 Bounding box 对于 Grid 网格的敏感度, 将目标中心点相对 Grid 网格左上角偏移量从原来的 (0, 1) 缩放到 (−0.5, 1.5).

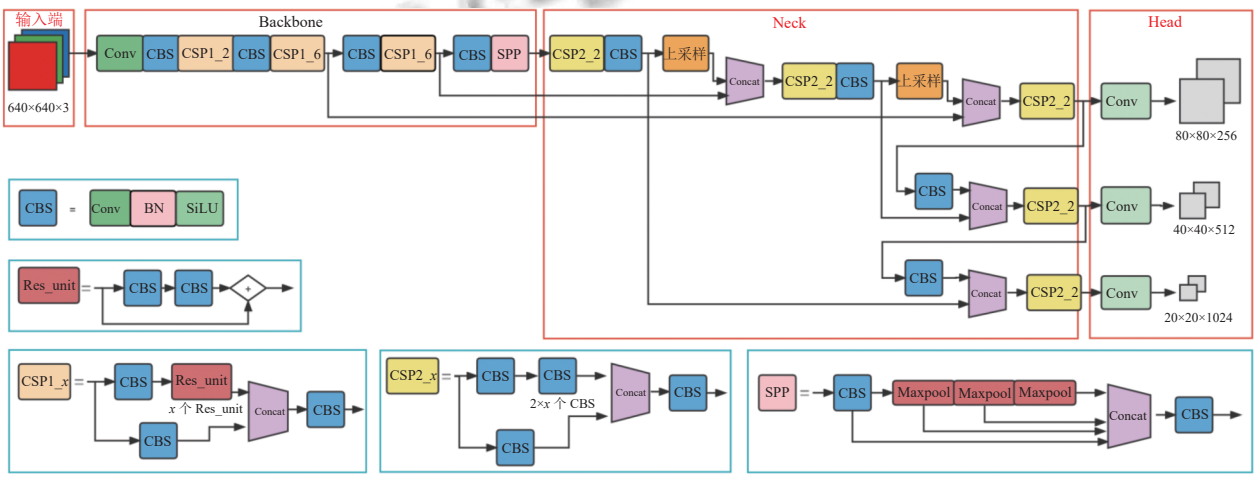


图 1 YOLOv5 网络结构

1.2 增加基于 GhostNet 的轻量化网络设计

传统的卷积网络存在特征信息冗余, 计算量大, 训练慢, 模型参数大等问题. 而 YOLOv5x 拥有较高的精度和召回率恰好得益于大量堆叠的卷积网络, 整个网络包括 567 层卷积, 参数量达到 86 231 272, 超大的网络结构对于硬件资源是一种挑战, 经常会超出内存限制, 在设备限制下, 模型常无法正常运行或只能在很低的 batch-size 下运行. 交通图像分辨率高、尺寸大, 如果使用传统卷积网络, 对硬件资源的要求极高, 势必需要减少模型的计算量和参数量.

GhostNet 是华为诺亚方舟实验室在 2020 年提出的轻量型网络结构, 在计算性能上超越谷歌开发的 MobileNetV3[30]. Ghost module 从特征图冗余问题出发, 利用特征图的相似性, 通过少量计算产生大量特征图. 由此 Ghost module 被设计为一种分阶段的卷积计算模块, 在少量的普通卷积得到的特征图基础上, 再进行一次线性卷积获取更多的特征图, 而新得到的特征图, 就被叫做之前特征图的“ghost”, 最后将两部分特征图拼接生成最终的特征图, 以此消除特征图冗余, 获得更加轻量的模型.

假设输入通道为 c, 特征图高和宽为 h 和 w, 输出数据的高度和宽度为 h' 和 w', 卷积核数量为 n, 卷积核大小为 k, 线性变换卷积核大小为 d, 变换数量为 s. 理论上, 使用 Ghost 卷积替换传统卷积的参数压缩比推算如式 (1) 所示:

$$r_c=\frac{n\cdot c\cdot k\cdot k}{\frac{n}{s}\cdot c\cdot k\cdot k+(s-1)\cdot\frac{n}{s}\cdot d\cdot d}\approx\frac{s\cdot c}{s+c-1}\approx s \tag{1}$$

加速比推算如式 (2) 所示:

$$\begin{aligned} r_s&=\frac{n\cdot h'\cdot w'\cdot c\cdot k\cdot k}{\frac{n}{s}\cdot h'\cdot w'\cdot c\cdot k\cdot k+(s-1)\cdot\frac{n}{s}\cdot h'\cdot w'\cdot d\cdot d}\\ &=\frac{c\cdot k\cdot k}{\frac{1}{s}\cdot c\cdot k\cdot k+\frac{(s-1)}{s}\cdot d\cdot d}\approx\frac{s\cdot c}{s+c-1}\approx s \end{aligned} \tag{2}$$

从式 (1) 和式 (2) 可以看出, 计算加速收益和参数压缩效果受变换数量影响, 即生成“ghost”特征图越多加速效果越好, 检测精度也会随之下降, 为平衡速度与精度一般将变换数量设置为 1/2.

利用 Ghost module 的优点, 进一步可以构建出适用于 YOLOv5 的 Ghost 结构如图 2 所示. GhostConv 首先使用原卷积一半大小的卷积生成一半的特征图, 再使用廉价计算对特征图加工, 生成另一半特征图, 最后通过 Concat 操作将两部分特征图拼接成完整的特征图, 其中廉价计算采用的是卷积核为 5, 步长为 1 的卷积. Ghost bottlenecks 的结构类似于 ResNet 的残差结构, 每个 Ghost bottlenecks 主要由两个 GhostConv 组成. 第 1 个 GhostConv 作为扩展层增加通道的数量. 第 2 个 GhostConv 的作用是减少输出特征图的通道数使其与输入通道数相匹配. 通过 shortcut 将两个 GhostConv 连接. 两个 GhostConv 之间的区别在于第 1 个 Ghost 后使用的是 ReLU 激活函数, 而后面的每层使用的都是批量归一化处理, 通过这样的结构方式使得模型在有效减少模型参数、计算量之余, 还能通过 GhostConv 对特征图进行优化, 提高模型检测效率.

CSPNet 结构是 YOLOv5 成功的关键, 利用跨阶段特征融合策略和截断梯度流技术增强不同网络层间学习特征的可变性, 从而减少冗余梯度信息的影响, 增强网络学习能力. 一方面, CSPNet 在减少计算量、减少内存成本的同时, 优化网络检测精度, 所以在改进 YOLOv5 模型时, 并没有删除 CSPNet 的架构, 反而将 CSPNet 应用到 GhostNet 网络里. 另一方面, 原 GhostNet 网络采用 MobileNetV3 结构, 网络结构太深, 在 YOLOv5 中使用该结构多尺度地提取特征, 会使模型参数量和计算量增加, 即使使用了轻量级网络, 也没有达到最终轻量级设计的目的. 综合两方面考虑, 设计 C3Ghost 结构, 将一半的特征信息通过 Ghost bottlenecks 生成特征图, 另一部分只通过卷积、正则化和激活函数生成特征图, 再将两部分特征图拼接, 使得梯度组合的差异最大化, 并减少大量梯度信息.

1.3 增加注意力机制

一方面, Ghost 模块使用的廉价线性变换是 depth-wise convolution (分组卷积), 分组卷积消除了通道间的相关性, 使得当前通道特征仅与自己相关, 虽然降低了参数量、计算量, 但模型对全局特征的提取减少. 另一方面, 在路测相机的图像中, 大面积的覆盖区域存在许多干扰因素, 我们希望将检测目标锁定在行驶的车辆上. 利用注意力机制提取感兴趣区域, 帮助 YOLOv5 抵御混淆信息, 并将算力集中在重要的特征上. 从以上两方面考虑, 需在模型中增加注意力机制, 在节约算力的同时, 提高模型检测精度.

受 Vision Transformer 的启发, 在文本中多头注意力可以获取上下文的信息, 将多头注意力机制用于图片可以获取全局信息, 即当需要识别一辆汽车时, 可以注意到有哪些块与汽车相关. 在主干网络中添加 Transformer block, 其结构如图 3(b) 所示, 在特征图放入

Transformer layer 训练前, 先进行 reshape, 调整参数顺序和维度, 得到形状为 3 维的向量, 并分配给 q、k、v, 放入多头注意力中训练学习. 多头注意力是多个自注意力的组合, 自注意力机制数学表达如式 (3) 所示:

$$Attention(q,k,v)=Softmax\left(\frac{q\cdot k^{\mathrm{T}}}{\sqrt{k\cdot dim}}\right)\cdot v \tag{3}$$

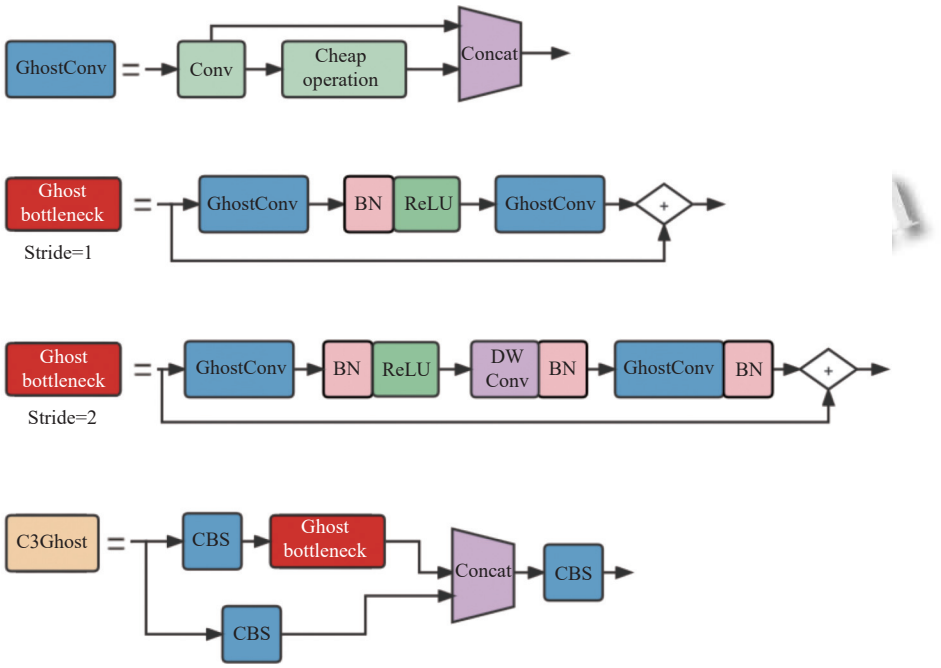


图 2 Ghost 结构示意图

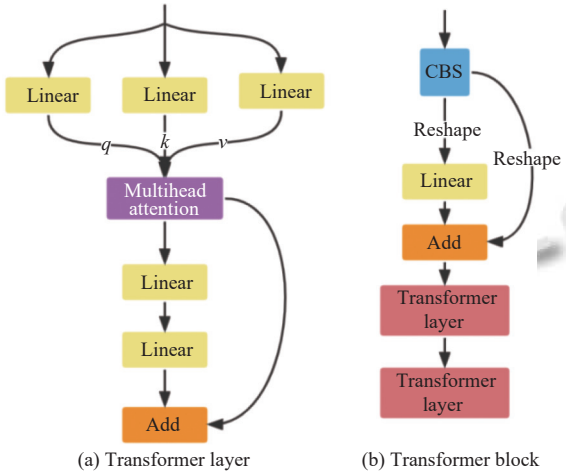


图 3 Transformer 结构示意图

多头注意力将模型分为多个部分, 形成多个子空间, 让模型关注不同方面的信息, 然后将不同的信息组合起来, 帮助当前节点不仅关注当前网格, 还能获取到周围环境信息. 同时, 多头注意力可以让每个自注意力头关注图像的不同特征, 挖掘特征表示的潜能.

在研究中显示, 自注意力机制常常会在数据量小的时候罢工甚至起负作用, 为了保证模型的稳定性, 减少 Transformer 使用, 在主干网络和特征融合网络间加入一层 Transformer 层, 而之后的注意力使用 CBAM 模块.

CBAM 模块是一种简单但有效的注意力模块, 特征图输入时, 会沿着两个独立的子模块 (通道注意力模块和空间注意力模块) 依次推断注意力权重, 然后将注意力权重与输入特征图相乘, 以进行自适应特征优化. 其中, 通道注意力模块将特征图进行最大池化和平均池化, 相当于把每个通道上所有特征图压缩为一个像素的特征, 再将池化后的输出经过多层感知器, 进行参数的学习, 并经过 *Softmax* 函数生成对于各个通道的重视程度, 因为特征图的每个通道都相当于一个特征检测器, 所以通道注意力可以让网络知道什么特征值得被关注. 在空间注意力中, 将经过通道注意力模块加权的结果沿着通道轴最大池化和平均池化, 把 n 个通道 $H\times W$ 的图像压缩为一个单通道的 $H\times W$ 的图像, 然后经过卷积层学习参数, 使用 *Softmax* 获取 $H\times W$ 上个像素点的重要程度, 即获取值得注意的位置. CBAM 可以作为轻量级模块集成到主流的 CNN 架构中, 以端到端方式对其进行训练, 并且 CBAM 模块可以减少无关信

息对检测识别的影响, 对减少误检率有很大帮助.

在 YOLOv5 架构中, 将 FPN (特征金字塔网络) 和 PAN (路径聚合网络) 结合进行多尺度特征融合, 其目的是将浅层网络的强位置信息和深层网络的强语义信息传递给其他网络层, 沿着这个思路把 CBAM 模块添加在 FPN 结构与 PAN 结构之间, 计算位置信息权重和语义信息权重, 关注重要的特征, 抑制不重要的信息. 整体注意力机制示意图如图 4 所示, 所有注意力模块嵌入基础 YOLOv5 网络, 并一起进行端到端训练.

1.4 基于 GhostNet 与注意力机制的 YOLOv5 模型

本文所提算法的整体结构如图 5 所示, 第 1 层 6×6 大小的卷积实际上与原 Focus 模块作用一样, 对输入的图像做切片操作, 例如 640×640×3 的图片输入, 会变成 320×320×64 的特征图. 所以第 1 层卷积没有变动, 其余所有卷积替换为 GhostConv, 所有 C3 结构用 C3Ghost 替代. 在 Backbone 网络提取特征的尾部添加 C3TR 模块, 该模块与 C3Ghost 一样借鉴 CSP 结构, 用 Transformer block 替换原本的 Bottleneck. 在 Neck 网络进行特征融合时, 使用 C3CBAM 模块加强重要特征的权重.

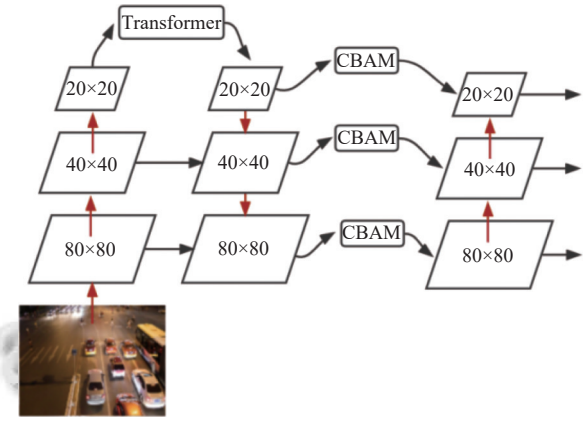


图 4 注意力机制示意图

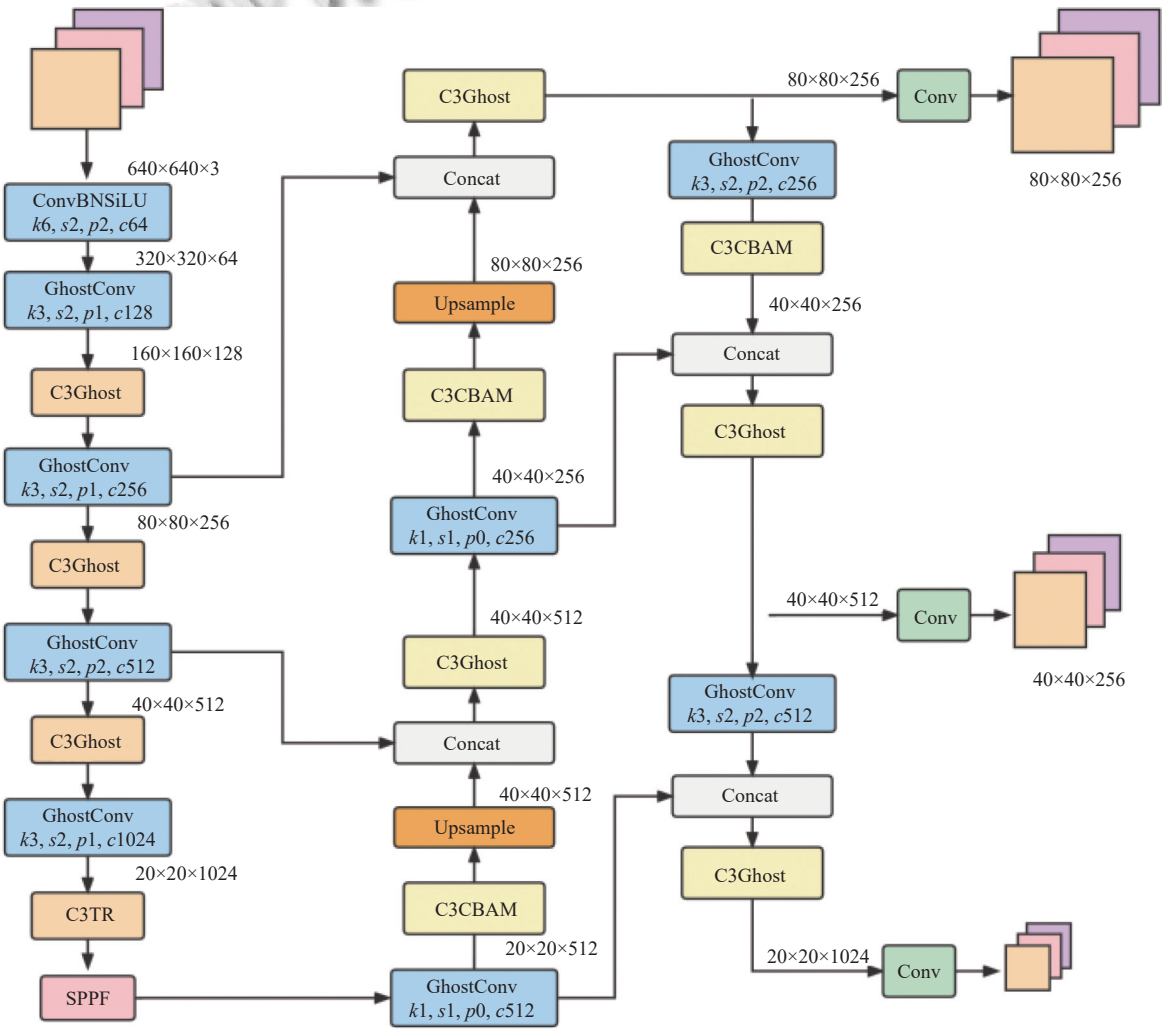


图 5 改进的 YOLOv5 模型结构示意图

2 实验与分析

2.1 数据集

本文使用 UA-DETRAC 数据集作为实验数据, UA-DETRAC 是一个具有挑战性的多目标车辆检测数据集. 该数据集包括在中国北京和天津的 24 个不同地点使用 Cannon EOS 550D 相机拍摄, 数据集中有 8 250 辆手动注释的车辆, 121 万个的目标框. 相较于其他车辆数据集, UA-DETRAC 采用大倾角俯视视角拍摄, 与基于车载设备平视视角不同, 更符合交通监管的场景, 且单张图像中有超过 30 辆车辆的场景, 拥有夜晚、雨天不同天气的图片, 包括十字路口、三叉路口等道路场景, 能够体现现实中复杂交通情景.

UA-DETRAC 数据集图片从视频中截取, 存在许多相似数据, 相邻图片差异很小, 使用完整的 82 090 张训练集, 训练速度非常慢, 且太多相似数据, 容易让模型过拟合. 为消除冗余的数据帧, 本文设计两个实验: 实验 1, 从数据集中抽取 8 000 张图片实验, 其中 6 462 张图片训练集、880 张验证集、406 张测试集, 主要用于消融实验, 测试每个模块对算法性能的影响; 实验 2, 从训练集的每个场景中随机抽取 40% 的图片, 其中, 20 521 张训练集, 12 312 张验证集, 从测试集每个场景抽取 20% 的图片, 一共 11 233 张作为测试集, 主要用于评估模型总体性能.

2.2 实验环境

本实验主要配置为系统: Linux 5.4 版本, 框架: PyTorch, CPU: Inter Xeon(R) CPU E5-2603 v4 @ 1.70 GHz x12 处理器, GPU: Nvidia GeForce GTX 1080 ti 10 GB, 内存: 128 GB, CUDA 版本 11.2.

实验设置输入图片大小为 640×640, 使用统一的超参数设置: 0.003 2 初始学习率, 0.12 周期学习率, 0.93 学习动量, 0.2 的 IoU 训练阈值, 批量大小为 16, 训练 100 epoch.

2.3 设计基于 K-means 与遗传算法的检测框获取方法

在 YOLO 系列算法中使用 anchor box 作为目标框的潜在候选框, anchor box 的选择将影响到 YOLO 算法的精度以及速度, 因此 anchor box 的长宽形状越接近真实的目标框越好. 由于 YOLO 网络的预测层包含 3 层感受野的信息, 每层感受野有 3 个锚框, 所以在模型训练前, 需要将车辆的大小聚类为 9 类, 替换 YOLOv5 初始 anchor box.

常用 K-means 聚类方法, 将从数据集中随机选取 K 个点作为初始聚类的中心点, 中心点为对于数据集中的每个样本 X_i, 计算它们到每个聚类中心点的距离, 并将它们划分到距离最小的聚类中心的类别中. 然后对于每个类别 i, 重新计算该类别的聚类中心, 重复以上步骤直到聚类中心的位置不变或达到迭代次数. 但是传统的 K-means 对于初始的聚类中心和样本输入顺序非常敏感, 容易陷入局部最优解.

本文采用基于遗传算法的 K-means 聚类, 将 K-means 聚类的局部寻优能力和遗传算法的全局寻优能力结合, 通过变异概率、种群迭代次数等因素找出最优解, 避免局部最优解的情况. 实现过程如下.

(1) 将 YOLO 基于长和宽的相对坐标转换为绝对坐标.

(2) 通过 K-means 聚类得到 n 个 anchors.

(3) 用遗传算法随机对 anchors 的长宽进行变异, 使用 anchor_fitness 方法计算变异后的适应度, 如果变异后效果变得更好就将变异后的结果赋值给 anchors, 如果变异后效果变差就跳过. 通过聚类和遗传算法计算 anchor 和 real bbox 的重合度, 使用遗传算法进化锚点. 设置变异次数 1 000 次, 得到新的预选框. 在 80×80 的网络上聚类出 anchor box 为 (20, 31), (70, 61), (95, 154). 在 40×40 网络上聚类结果为 (30, 43), (56, 91), (172, 162). 在 20×20 网络上聚类结果为 (43, 54), (102, 92), (129, 384).

2.4 模型性能评价指标

采用检测精度、召回率、平均精度、检测速度、模型大小作为模型性能的评价指标. 检测精度 (precision, P) 是模型预测目标中, 预测正确的比例, 计算公式为:

$$P = \frac{TP}{TP + FP} \tag{4}$$

召回率 (recall, R) 是所有真实目标中, 模型预测正确目标的比例, 计算公式为:

$$R = \frac{TP}{TP + FN} \tag{5}$$

平均精度 (mean average precision, mAP) 是所有类别预测精准度的平均值, 计算公式为:

$$AP = \int_0^1 P(R)dt \tag{6}$$

$$mAP=\frac{\sum_{n=0}^{N}AP_n}{N} \tag{7}$$

其中, 设置 IoU 为 0.5 时, TP (true positive) 表示预测框与真实标签 IoU 大于 0.5 的数量, FP (false positive) 表示 IoU 小于 0.5 的数量, FN (false negative) 表示没有检测到真实标签的数量. 实验中使用了两种 mAP, 分别是 mAP@0.5 和 mAP@.5:.95, mAP@0.5 表示 IoU 设定为 0.5 时, 所有类别的平均精确, mAP@.5:.95 表示 IoU 阈值在 (0.5, 0.95) 区间内, 步长为 0.05, 分别计算 mAP, 然后取平均值. 目标检测常用的是 mAP@0.5, 而使用 mAP@.5:.95 可以更加关注预测框位置的准确性.

检测速度常用两种指标, 一种是检测每张图片所需时间, 单位为 ms, 一种是每秒检测图片数量, 单位 FPS, 两者间可相互换算. 模型大小的评估指标采用计算量和参数量两个指标, 计算量是模型所需浮点运算次数, 用以衡量模型时间复杂度, 单位 GFLOPs, 参数量是模型参数的数量总和, 用以衡量模型空间复杂度, 单位为 M.

2.5 损失函数

损失函数通过衡量模型预计结果和实际数据的差距, 引导模型训练方向. 本文损失函数由 3 部分组成, 分别为 box_loss (定位损失)、obj_loss (分类损失) 和 cls_loss (置信度损失), 如式 (8) 所示: t_p, t_{gt} 为预测向量与真实向量; K, S^2, B 分别表示特征图数量、网格数量以及每个网格上预测框的数量; a_*表示对应项的权重, 训练在 hyp.scratch-low.yaml 上微调, a_{box}=0.05, a_{cls}=0.3, a_{obj}=0.7; a_k^{balance}是用于平衡每个尺度输出特征图的权重; II_{kij} 表示对应预测框是否为正样本, YOLOv5 中 3 部分的损失函数均是通过匹配的正样本计算, 没有正样本的特征图不参与计算;

$$L_{\text{total}}(t_p,t_{gt})=\sum_{k=0}^{K}\left[a_k^{\text{balance}}\left(a_{\text{box}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\text{II}_{kij}^{\text{obj}}L_{CIoU}+a_{\text{obj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\text{II}_{kij}^{\text{obj}}L_{\text{obj}}+a_{\text{cls}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\text{II}_{kij}^{\text{obj}}L_{\text{cls}}\right)\right] \tag{8}$$

定位损失采用 $CIoU\ loss$, 如式 (9) 所示: IoU 为图像重叠面积; $\rho^2(b,b^{gt})$表示预测框和目标框的中心点的欧式距离, c^2表示对角线距离; aV表示预测框和目标框的长宽比. $CIoU\ loss$ 将目标框回归函数中 3 个重要几何因素 (重叠面积、中心点距离, 长宽比) 包括在定位损失中.

$$L_{CIoU}=1-IoU+\frac{r^2(b,b^{gt})}{c^2}+aV \tag{9}$$

$$a=\begin{cases}0, & \text{if } IoU<0.5\\ \dfrac{V}{(1-IoU)+V}, & \text{if } IoU\geqslant 0.5\end{cases} \tag{10}$$

$$V=\frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}}-\arctan\frac{w}{h}\right)^2 \tag{11}$$

分类损失和置信度损失都采用基于 $Sigmoid$ 的二进制交叉熵函数 (BCEWithLogitsLoss), 整体公式如式 (12) 和式 (13) 所示:

$$L_{\text{obj}}(p_0,p_{IoU})=BCE_{\text{obj}}^{Sigmoid}\left(p_0,p_{IoU};w_{\text{obj}}\right) \tag{12}$$

$$L_{\text{cls}}\left(c_p,c_{gt}\right)=BCE_{\text{obj}}^{Sigmoid}\left(c_p,c_{IoU};w_{\text{cls}}\right) \tag{13}$$

2.6 实验结果分析

2.6.1 GhostNet 特征图

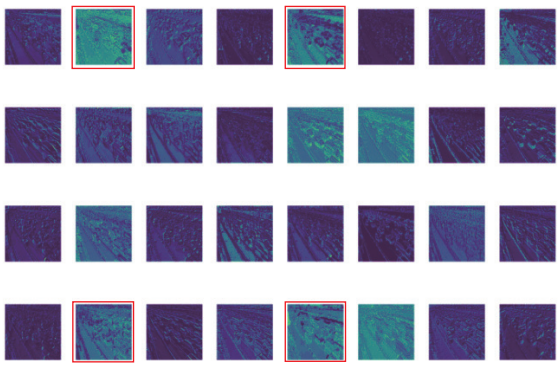
在 YOLOv5s 中使用 Ghost 模块进行卷积操作, 模型的参数量由 7 071 633 下降到 3 692 633, 模型参数压缩了 52%, 计算量由 16.5 GFLOPs 下降到 8.1 GFLOPs, 计算加速比为 49%. Ghost 轻量化的效果和式 (1)、式 (2) 得出的结论相差不大, 使用 Ghost 生成 50% 特征图, 加速比和压缩率接近 50%.

Ghost 采用廉价操作生成一部分的特征图, 那么将 YOLOv5 的卷积全替换成 Ghost 卷积, 特征信息是否会因此丢失, 带着这样的疑问, 将 YOLOv5s 和 Ghost-YOLOv5 提取的特征可视化, 如图 6 所示, 特征图分别为经过第 1 个 C3 结构和 C3Ghost 结构的结果, 大体上两份特征提取的结果相近, 并且在方块圈出来的特征图里, Ghost-YOLOv5 将车辆线条提取的更加明显, 车辆整体的轮廓保留在特征图中. 之后, 使用验证集得到 YOLOv5s 检测精度为 0.970 3, 召回率为 0.958 0, Ghost-YOLOv5 检测精度为 0.967 9, 召回率为 0.938 5, 精度下降 0.24%, 召回率下降 2.94%, 得出结论 Ghost 在帮助模型压缩参数和减少计算量上十分有效, 但会损失一些精准度.

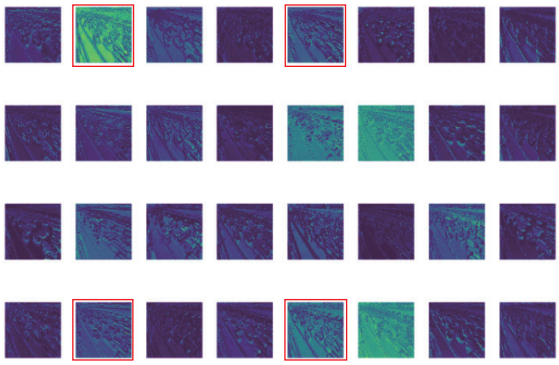
2.6.2 模块有效性评价

为了分析各模型组合对模型影响, 设计消融实验, 训练采用统一超参数, 训练 100 epoch, 取 mAP 最高的结果保存, 结果如表 1 所示, Ghost 轻量化效果很好, 使

用 Ghost 的模型参数量最低, 仅有 3.5 M, 但是检测准确率和召回率下降. CBAM 可以提升模型的精确度和召回率, *mAP*@.5:.95 指标比 YOLOv5s 提升了 1.82%, 说明模型边框回归的更好, 作为轻量级模块加入模型, 参数量仅增加了 1 M. 值得注意的是, 将 Ghost 和 CBAM 同时使用, 对比只使用 CBAM 注意力的模型检测效果更好, 并且参数量更少. CBAM 帮助 Ghost 更好的选择重要的特征, 排除干扰信息, 建立通道间相关性, 对比 YOLOv5s, *P* 提高 1%, *R* 提高 1.1%, *mAP*@.5:.95 提升 2.68%, 参数量减少 26%, 在模型检测性能提高的同时, 降低模型对硬件资源的需求. 本文算法将 Ghost、CBAM、Transformer 这 3 种思想融入 YOLOv5s, 在消融实验中, 为了方便训练, 只抽取了 8 000 张图片, 但数据集太小使得 Transformer 模块并没有良好表现, 对比只加入 Ghost 和 CBAM 的模型, 性能有轻微下降.



(a) YOLOv5 提取特征图



(b) Ghost-YOLOv5 提取特征图

图 6 注意力机制示意图

2.6.3 综合性能评价

为进一步研究本文算法的性能, 数据集从 8 000 张扩充到 44 066 张, 对原始的 YOLO 系列检测算法和改进算法做性能对比分析, 结果如表 2 所示, 本文算法对标 YOLOv5x, *mAP*@0.5 由 0.976 3 提升到 0.986 8, *P* 由 0.933 5 提升到 0.976 0, *R* 由 0.954 4 提升到 0.965 2, 同时参数量从 86.7 M 减少到 47 M, 压缩了 46%, 速度由 37 FPS 提高到 65 FPS, 提升了 43%. 使用 11 233 张测试集检验模型泛化性, 本文的算法在测试集上 *mAP*@0.5 比 YOLOv5x 高出 2.4%. 对比 YOLOv5x, 本文算法在检测速度、精度、模型大小、泛化性能上都得到了优化.

表 1 消融实验结果对比

算法	*P*	*R*	*mAP*@.5:.95	参数量 (M)
YOLOv5s	0.970 3	0.958 0	0.856 9	7.2
+Ghost	0.967 9	0.938 5	0.836 6	**3.5**
+CBAM	0.976 2	0.969 7	0.875 1	8.2
+Transformer	0.973 1	0.956 7	0.866 1	7.2
+Ghost&CBAM	**0.980 8**	0.969 9	**0.883 7**	5.0
本文算法	0.975 5	**0.974 9**	0.880 8	5.6

表 2 网络模型性能对比表

算法	*P*	*R*	*mAP*@0.5	参数量 (M)	速度 (FPS)	测试集 *mAP*@0.5
YOLOv3-SPP	0.938 1	0.961 7	0.977 5	60	**138**	0.686
YOLOv5s	0.924 7	0.953 2	0.970 6	**7.2**	81	0.685
YOLOv5x	0.933 6	0.954 5	0.976 3	86.7	37	0.715
本文算法	**0.976 0**	**0.965 2**	**0.986 8**	47	65	**0.751**

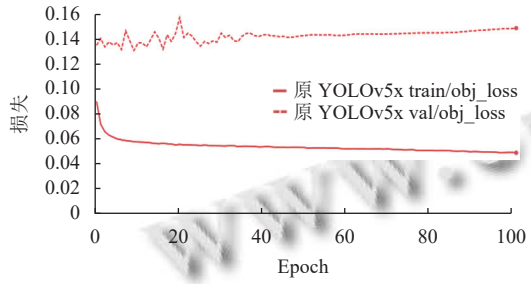
对比 YOLO 系列优秀算法, 本文算法在检测精度、召回率、平均精度优于以往算法, 并且参数量不高, 对硬件需求不大, 在实时性方面一般将 30 FPS 作为标准, 一秒检测图片数量大于 30 张表示算法具备实时性, 改进后的模型具有 65 FPS 的速度, 超过标准 116%.

2.6.4 DropBlock 正则化

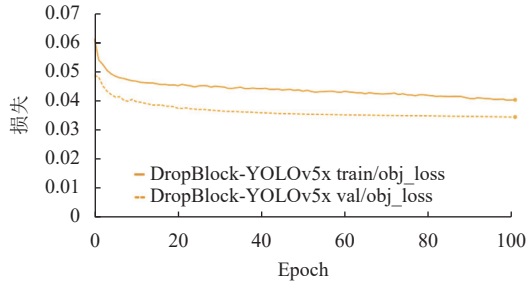
UA-DETRAC 数据集采集使用 Cannon EOS 550D 相机以固定的路测视角拍摄, 并且视频拍摄时间处于连续时间段, 训练出的模型鲁棒性低且容易过拟合. 如图 7(a) 所示, 模型在训练集上, 分类损失平滑地收敛; 但在验证集上, 分类损失在训练前期处于抖动状态, 在训练后期也没能很好收敛. 同时训练出的模型在训练集上 *mAP* 达到 97.63%, 在测试集上 *mAP* 只有 71.5%, 模型总在训练集上拥有良好表现, 可能过于依赖训练场景的某些特征, 导致模型泛化性不足.

为增强模型泛化性, 并使得分类损失较好的收敛. 本文在训练阶段先使用标签平滑技术和调整置信度损失和目标框损失权重, 但模型过拟合问题并没有

得到很好的改善. 在 Detect 层利用卷积处理特征图的过程中使用 DropBlock 模块, 设置屏蔽块大小为 3, 屏蔽块的数量占特征图的 30%. 即模型在处理特征图时, 随机屏蔽 30% 相邻连续的特征区域块, 避免模型过于依赖部分特征, 获得更好的泛化性. 由图 7(b) 可以看出模型在验证集上的损失也变得更加平滑, 同时在测试集上 *mAP* 也提升到 74%. 说明在 YOLOv5 检测头加上 DropBlock 正则化, 能够帮助模型更好地训练, 提高在验证集和测试集上的检测性能, 增强模型泛化能力.



(a) 原 YOLOv5x 分类损失收敛图



(b) 使用 DropBlock的 YOLOv5x 分类损失收敛图

图 7 训练集和测试集上的分类损失收敛图

2.7 检测结果对比

为了体现本文算法的有效性, 从测试集中选取一些图像进行检测. 检测效果如图 8 和图 9 所示, 在第 1 组检测图片中, 图中车辆超过 30 辆, 包括小轿车和公交车, 存在车辆遮挡、车辆拥挤等困难, 属于复杂交通场景. 在 YOLOv5 算法检测中出现许多漏检情况, 还有将中间的公交车识别成小轿车的误检情况, 使用本文改进的 YOLOv5 算法检测, 能够大大降低漏检情况, 并且没有出现误检. 在第 2 组检测图片中, 车辆处于低光照强度环境且存在车灯干扰情况, 从图 8(a) 和图 9(a) 中可以看到 YOLOv5 算法未检测出左上角被公交车遮挡的车辆和右下角刚刚出现的车辆, 使用本文算法能准确地将它们检测出来. 上述检测效果表明, 本文提出的模型能够实现对复杂交通场景下车辆的有效检测.

(a) 原 YOLOv5 检测结果

(b) 改进 YOLOv5 检测结果

图 8 复杂交通场景

(a) 原 YOLOv5 检测结果

(b) 改进 YOLOv5 检测结果

图 9 夜晚低光照场景

3 结论与展望

为解决车辆检测在复杂交通场景中召回率和精确率较低、模型参数量大、检测速度慢等问题, 本文提出基于 GhostNet 与注意力机制的 YOLOv5 车辆检测

模型. 采用基于遗传算法的 K-means 聚类方法, 得到适用于车辆检测的预选框; 通过加入 DropBlock, 提高模型泛化性; 将传统卷积替换为 Ghost 卷积并构建 C3Ghost 模块代替原 CSP 结构, 减少模型参数, 减少计算成本, 并在主干网络和检测头添加 Transformer block 和 CBAM 注意力模块, 加强模型特征提取能力和检测能力. 实验结果表明, 在 YOLOv5x 基础上改进的算法, 车辆检测精度达到 97.57%, 召回率到达 96.48%, 较 YOLOv5x 模型检测精度提高 4.21%, 召回率提升 1.03%, 参数量减少 35 M, 速度提高一倍, 泛化性提高, 本文模型可以在拥挤场景下检测出重叠目标, 低光照场景下识别出更多交通目标, 减少漏检、误检情况. 目前, 模型泛化性依旧不强, 算法对预测未知数据的能力不够, 在训练集上有很好的表现, 但是在测试集上表现不佳, 除了扩充数据集、正则化以及调参外, 之后可以使用蒙特卡罗 Dropout (MC Dropout) 和贝叶斯神经网络对模型不确定性进行研究, 利用不确定性判断预测的可信程度, 从而避免一些错误的预测.

参考文献

1 Hadi RA, Sulong G, George LE. Vehicle detection and tracking techniques: A concise review. arXiv:1410.5894, 2014.

2 Zivkovic Z, van der Heijden F. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters, 2006, 27(7): 773–780. [doi: 10.1016/j.patrec.2005.11.005]

3 Godbehere AB, Matsukawa A, Goldberg K. Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. Proceedings of the 2012 American Control Conference. Montreal: IEEE, 2012. 4305–4312.

4 Farnebäck G. Two-frame motion estimation based on polynomial expansion. Proceedings of the 13th Scandinavian Conference on Image Analysis. Halmstad: Springer, 2003. 363–370.

5 Lucas BD, Kanade T. An iterative image registration technique with an application to stereo vision. Proceedings of the 7th International Joint Conference on Artificial Intelligence. Vancouver: ACM, 1981. 674–679.

6 Dalal N, Triggs B. Histograms of oriented gradients for human detection. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego: IEEE, 2005. 886–893.

7 Ma XX, Grimson WEL. Edge-based rich representation for vehicle classification. Proceedings of the 10th IEEE International Conference on Computer Vision. Beijing: IEEE, 2005. 1185–1192.

8 Papageorgiou CP, Oren M, Poggio T. A general framework for object detection. Proceedings of the 6th International Conference on Computer Vision. Bombay: IEEE, 1998. 555–562.

9 Kazemi FM, Samadi S, Poorreza HR, *et al*. Vehicle recognition using curvelet transform and SVM. Proceedings of the 4th International Conference on Information Technology. Las Vegas: IEEE, 2007. 516–521.

10 Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 1997, 55(1): 119–139. [doi: 10.1006/jcss.1997.1504]

11 Girshick R. Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision. Santiago: IEEE, 2015. 1440–1448.

12 陶颖军. 基于 OpenCV 的人脸识别应用. 计算机系统应用, 2012, 21(3): 220–223. [doi: 10.3969/j.issn.1003-3254.2012.03.051]

13 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]

14 黄继鹏, 史颖欢, 高阳. 面向小目标的多尺度 Faster-RCNN 检测算法. 计算机研究与发展, 2019, 56(2): 319–327. [doi: 10.7544/issn1000-1239.2019.20170749]

15 陈飞, 章东平. 基于多尺度特征融合的 Faster-RCNN 道路目标检测. 中国计量大学学报, 2018, 29(4): 393–397. [doi: 10.3969/j.issn.2096-2835.2018.04.008]

16 Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 6517–6525.

17 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.

18 孟乔. 面向复杂场景的车辆检测跟踪及行为分析关键技术研究 [博士学位论文]. 西安: 长安大学, 2021.

19 张新宇, 丁胜, 杨治佩. 基于改进注意力机制的交通标志检测算法. 计算机应用, 2022, 42(8): 2378–2385. [doi: 10.11

772/j.issn.1001-9081.2021061005]

20 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. Seattle: IEEE, 2020. 1577–1586.

21 Jocher G. YOLOv5. https://github.com/ultralytics/yolov5. (2020-08-09).

22 Wang CY, Liao HYM, Wu YH, *et al*. CSPNet: A new backbone that can enhance learning capability of CNN. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Seattle: IEEE, 2020. 1571–1580.

23 Dosovitskiy A, Beyer L, Kolesnikov A, *et al*. An image is worth 16×16 words: Transformers for image recognition at scale. Proceedings of the 9th International Conference on Learning Representations. OpenReview.net, 2021. 1–10.

24 Woo S, Park J, Lee JY, *et al*. CBAM: Convolutional block attention module. Proceedings of the 15th European Conference on Computer Vision. Munich: Springer, 2018. 3–19.

25 Wen LY, Du DW, Cai ZW, *et al*. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. Computer Vision and Image Understanding, 2020, 193: 102907. [doi: 10.1016/j.cviu.2020.102907]

26 赖玉霞, 刘建平, 杨国兴. 基于遗传算法的 K 均值聚类分析. 计算机工程, 2008, 34(20): 200–202. [doi: 10.3969/j.issn.1000-3428.2008.20.073]

27 Ghiasi G, Lin TY, Le QV. DropBlock: A regularization method for convolutional networks. Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal: ACM, 2018. 10750–10760.

28 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. Honolulu: IEEE, 2017. 936–944.

29 Liu S, Qi L, Qin HF, *et al*. Path aggregation network for instance segmentation. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8759–8768.

30 Howard A, Sandler M, Chen B, *et al*. Searching for MobileNetV3. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 1314–1324.

(校对责编: 孙君艳)