

基于检测框匹配的异常停车检测方法

李莹莹¹, 李海芳^{1*}, 宋瑞霞², 刘战东¹, 李克¹, 丁男¹

(1. 新疆师范大学 计算机科学技术学院, 新疆 乌鲁木齐 830054; 2. 新疆师范大学 图书馆, 新疆 乌鲁木齐 830017)

摘要: 文章提出了一种车辆检测网络模型,旨在更好地检测小型车辆目标,提高目标检测的准确性,从而实现更具鲁棒性的异常检测。同时,文章还探讨了一种基于检测框匹配的异常停车检测方法,以期提升异常检测的准确性。为了优化异常检测方法,本研究引入车辆检测网络模型于检测车辆目标,并结合目标检测和跟踪方法生成道路掩码。通过综合应用,进一步完善异常检测器,实现对异常事件开始时间和结束时间的准确获取。

关键词: 计算机视觉; 目标检测; 异常检测; 背景建模; 道路掩码

中图分类号: TP391 **文献标识码:** A **文章编号:** 1008-9659(2025)02-0008-12

近年来,随着城市化进程加快,机动车数量急剧增加,交通拥堵和事故发生频率上升,给城市管理和公共安全带来挑战。同时,随着我国经济持续增长,交通监控摄像头已经成为城市道路、交叉路口和重要场所的常见设施。这些摄像头每毫秒产生海量数据,梳理数据并进行检索是一项艰巨的任务。面对如此庞大的数据量,通过人工手段进行筛查是不现实的。

随着深度学习的发展,图像处理、模式识别、深度学习等技术为获取视频中有效信息提供了技术支持,智慧交通系统随之应运而生,其不仅能提高交通管理的效率和精准性,还能有效解决交通安全和交通拥堵等城市交通问题。智慧交通系统通过先进的技术,如人工智能、大数据分析和物联网等,实现对交通情况的实时监测、分析和调控,为城市交通运行提供更智能、高效的管理方式。

在智慧交通系统中,交通异常检测是不可或缺的一部分。交通异常检测任务涵盖了各种可能的异常行为,例如车辆违规变道、车辆超速行驶、车辆逆行、车辆碰撞等。文章研究车辆异常停车情况,例如车辆碰撞引起的停车以及车辆熄火等异常停车情况。异常停车事件会极大地降低交通效率,车辆异常停车在交通异常行为中占比较高。通过及时识别和处理交通异常情况,可以有效提高交通系统的运行效率和城市交通的通畅性,从而改善居民的出行体验和城市的交通安全,所以异常停车检测研究极具研究价值。

当前,大多数异常停车检测方法均基于深度学习,其依赖于目标检测的结果。尽管目标检测在精度和速度方面取得了显著进展,但在检测小目标方面仍存在一定挑战,特别是针对监控视频中远距离目标的情况。因此,目标检测研究对于异常停车检测非常必要。异常检测任务追求实时性,越早发现对于后续处理越有利,然而大多数方法都无法做到对异常情况进行及时检测,所以异常检测的研究还有待进一步探索。

1 相关工作

一般使用统计模型来获取正常事件的相关信息。隐马尔科夫模型是目前最流行的统计模型,用于检测

[收稿日期] 2024-04-11

[修回日期] 2024-06-01

[基金项目] 国家自然科学基金项目(62162061;62262066);新疆维吾尔自治区自然科学基金项目(2021D01E20);新疆维吾尔自治区重点研发计划项目(2022B01007-1;2022B01007-3);新疆师范大学博士启动基金项目(XJNBS2115);新疆师范大学青年拔尖人才项目(XJNUQB2022-21)。

[作者简介] 李莹莹(1999-),女,硕士研究生,主要从事计算机视觉方面研究,E-mail:liyy0405@163.com.

* [通讯作者] 李海芳(1986-),女,讲师,主要从事智慧交通、深度学习和计算机视觉方面研究,E-mail:951624998@qq.com.

场景中的异常事件^[1]。它主要通过建立局部时空运动模式行为的统计框架来检测极度拥挤场景中的异常事件,关注建模拥挤场景中的局部运动模式,以捕获潜在结构并检测异常活动作为统计偏离。一种主要的算法是使用二进制分类器,如支持向量机,这些方法使用从视频中提取的描述符来训练分类器^[2-5]。另一种主要的研究方法是在稀疏重建框架中执行异常事件检测。该方法使用从正常训练特征中学习的稀疏字典来表示测试视频,估计稀疏重建成本(重建误差)并对其进行阈值处理以检测异常^[6-10]。

随着深度学习的发展,深度学习技术特别是卷积神经网络为各种计算机视觉任务提供了有效的解决方案,如对象检测、图像分类、动作识别等。基于深度学习的异常检测方法主要分为:基于检测跟踪的异常检测方法和基于目标检测的异常检测方法。基于目标检测的异常检测方法首先会对视频帧进行背景建模来消除移动车辆,随后选择一个目标检测器来进行检测,最后进行异常分析^[11-18]。Li等人^[19]基于检测跟踪的异常检测方法,提出了一种多粒度跟踪方法,该方法结合了盒级跟踪分支和像素级跟踪分支来分析不同粒度级别的候选异常车辆。其首先分别对前景和背景进行背景建模,随后构建道路掩码,通过Faster R-CNN检测器对视频帧进行检测,通过像素级跟踪模块对背景信息、掩码信息、检测结果进行分析,获得异常时间,最后通过回溯算法获得更精确的异常时间,基于跟踪的大多数异常检测方法都是这样的流程。

基于深度学习的两种异常检测方法各有优缺点。在检测性能方面,基于检测跟踪的异常检测方法能够提供更准确的异常检测结果,因此更具优势。然而,就实际应用和部署而言,基于检测的异常检测方法更具可行性和灵活性。基于检测跟踪的异常检测方法需要在监控系统中实时跟踪目标,并对其行为进行分析,这在计算资源和算法复杂度方面提出了更高的要求。尽管其在检测性能上表现出色,但在实际应用中可能会受到资源限制和系统复杂度的影响,导致部署难度增加。相比之下,基于检测的异常检测方法更简单直接。这使得基于检测的方法更易于部署和在实际场景中,尤其适用于资源受限或需要快速部署的情况。

文章提出的异常检测方法基于目标检测范式,将文章提出的车辆检测网络模型用于检测车辆目标以及背景帧中的车辆目标,通过构建道路掩码消除停车场等噪声项,通过异常检测器分析异常,获得异常的开始时间和结束时间。

2 研究方法

本研究旨在利用目标检测模型扫描交通视频背景帧,检测可能出现的交通异常事件。首先,通过检测模型识别潜在异常情况,随后对这些候选异常进行滤波处理,并与前景检测结果进行匹配,以推测异常事件的开始时间和结束时间。文章提出的异常停车检测方法框架如图1所示。

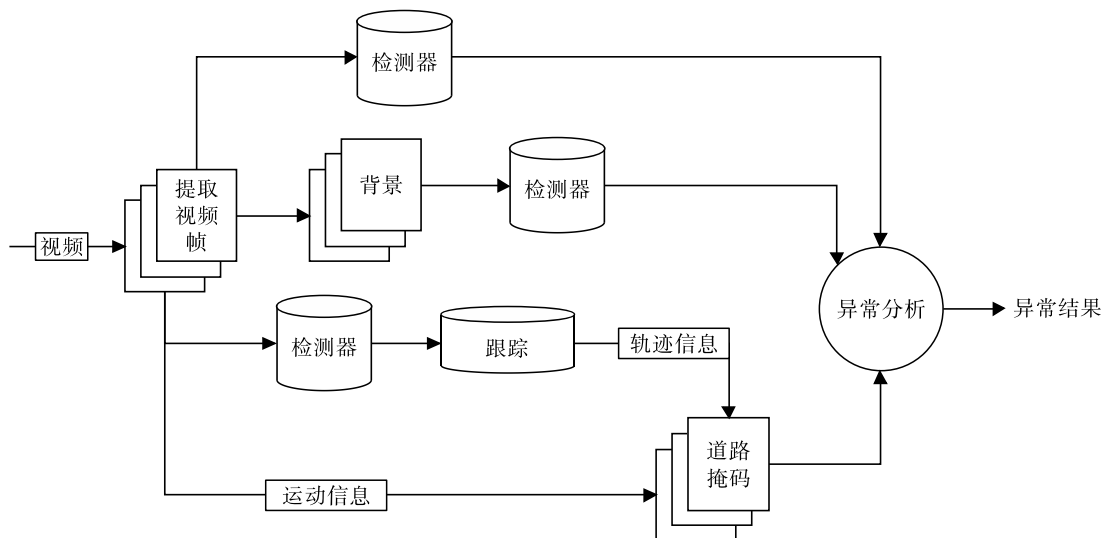


图1 异常检测框架图

2.1 背景提取

本研究利用高斯混合模型技术从视频中提取背景信息。该方法具有很强的适应能力,可以有效克服光照变化、时间噪声和图像抖动等问题,在几帧内获得稳定的背景。为了创建背景掩码,将生成的背景每分钟

平均一次,这个过程会持续至视频结束,每15分钟视频将产生14个1分钟的背景掩码。在背景创建过程中,舍弃最后一分钟背景。此外,文章每分钟对背景帧进行采样,提高计算效率。基于每分钟采样,检测器只需在每个视频中运行一次,而不是对每一帧背景视频都进行检测,这有助于减少计算量并加快处理速度。

2.2 道路掩码

通常情况下,主干道长时间停放的车辆将被视为异常情况,而停放在停车场的车辆则不被认定为异常。为了消除路边和停车场停车的影响,必须将主干道之外的区域分离出来,仅考虑主干道上的车辆。通过构建道路掩码,可以排除停车场等主干道外区域的干扰,以应对路面场景的复杂性和异常情况偏离主干道的情况。

本研究通过分析连续两帧图像之间的差异来构建异常掩码。具体做法是计算相邻帧图像的像素差异,根据设定的阈值判断差异是否超过阈值。若差异超过阈值,则认为存在移动物体,并将该区域保留。但为了应对摄像头抖动和场景变化等因素,将设置另外一个阈值,若某帧的变化量超过该阈值,则放弃该结果以减少误报。最终,将视频中所有变化区域相加生成最终的运动掩码。尽管通过连续两帧图像差异构建的道路掩码基本覆盖主干道路,但并非总能达到预期效果。

因此,本研究在原有构建道路掩码的基础上,增加了通过分析车辆轨迹来生成道路掩码的策略。首先,利用多目标跟踪算法DeepSORT来追踪车辆,获取车辆的运动轨迹。针对每个车辆轨迹,设定两个阈值:轨迹长度阈值 c 和轨迹行进距离阈值 d 。若轨迹长度小于阈值 c 或者轨迹的行进距离小于阈值 d ,则将该轨迹视为误检或者辅助道路上的轨迹,不考虑其中的车辆检测结果。处理完车辆轨迹后,将每个轨迹对应的车辆检测结果进行累计,从而获得基于车辆轨迹的掩码。此外,对掩码进行处理,去除面积较小的连通区域,以消除辅助道路等干扰项,从而获得更准确的异常检测结果。

最后,将上述两个掩码取交集,得到最终的道路掩码。基于车辆轨迹的方法能够更准确地呈现车辆在道路上的运动情况,提高生成道路掩码的准确性和稳定性。图2(a)展示了采用第一种方法生成的道路掩码,图2(b)展示了采用第二种方法生成的道路掩码,图2(c)是融合后的结果,图2(d)是视频的背景图片。通过比较可见,仅使用第一种方法生成的道路掩码质量较低,融合第二种方法后,掩码质量明显提升。然而,有时基于第二种方法生成的道路掩码可能质量较差。在这种情况下,可对加入第一种方法生成的掩码进行补充,生成更高质量的道路掩码。此融合方法能够有效提高道路掩码质量,为后续异常检测和分析提供更准确基础。综合利用不同方法生成的道路掩码,能更全面地捕捉道路特征,进而提高异常检测的准确性和鲁棒性。

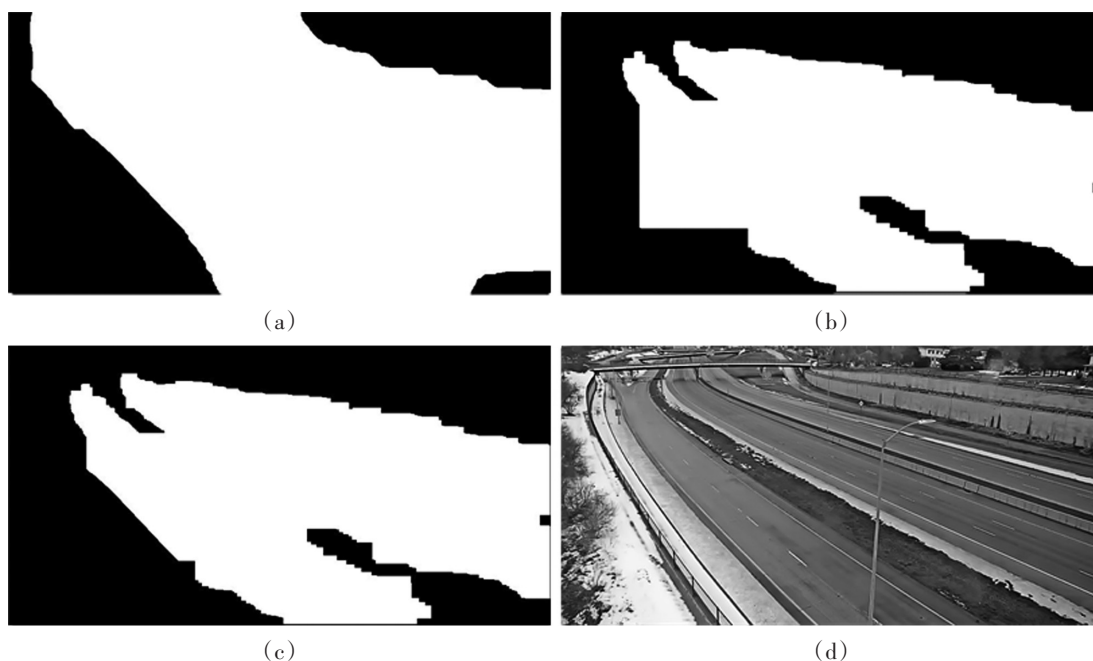


图2 道路掩码

2.3 车辆检测网络模型

YOLOv8目标检测模型的网络结构主要由三个关键部分组成:主干网络、网络部分和检测头部分。文章提出的改进模型对YOLOv8目标检测模型进行了修改和优化。具体来说,是对网络部分和主干网络部分进行针对性的改进。在网络部分,采用分组通道交换卷积(Group Shuffle Convolution, GSConv)模块来替代原来的卷积模块。同时,重新设计GS Bottleneck模块和跨阶段部分网络模块VOV-GSCSP,然后用VOV-GSCSP模块替换第三个C2f模块。这些修改旨在提升特征提取和信息融合的效率,以更好地适应目标检测任务的需求。在主干网络部分,引入轻量卷积模块LightConv,将其用于替换第三个卷积模块。这一改进旨在降低模型的数量和计算复杂度,从而提高模型的轻量化和高效性。图3展示了改进模型的网络结构相比于YOLOv8目标检测模型的网络结构图。

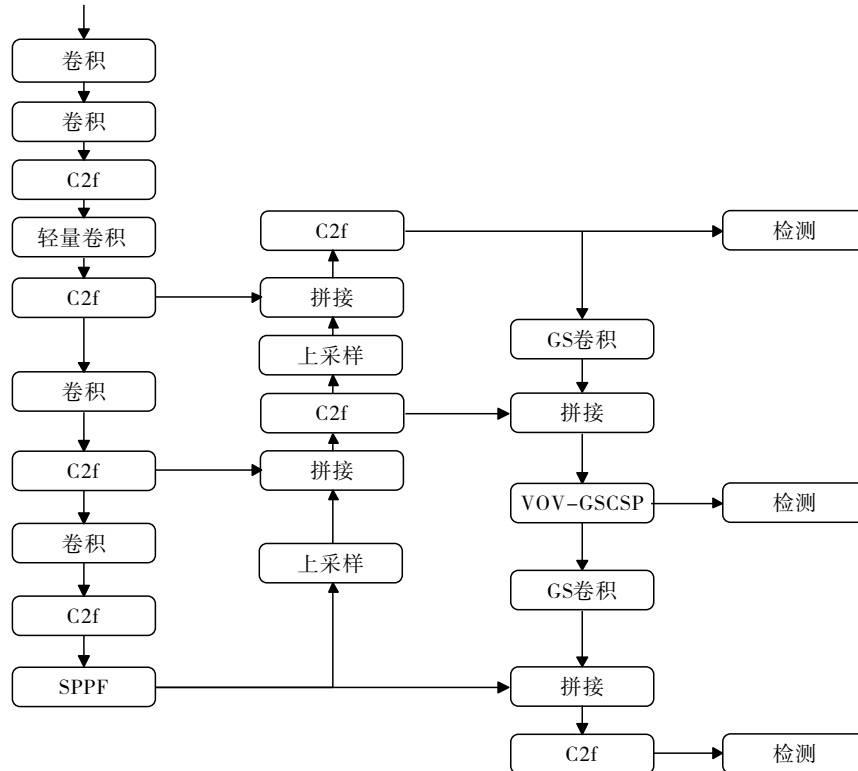


图3 车辆检测网络模型架构图

分组通道交换卷积,即GSConv^[20],是一种新型的轻量级卷积方法,旨在使深度可分离卷积与传统卷积的效果更接近、高效。其模块结构如图4所示。GSConv通过在深度可分离卷积中引入混合卷积,结合传统卷积、深度可分离卷积和通道重排操作,从而在减少计算成本的同时保持模型的准确性。GS Bottleneck是一种基于GSConv的增强模块,用于提高特征的非线性表达和信息的复用。VOV-GSCSP模块是利用一次性聚合方法设计的跨阶段部分网络模块,用于在不同阶段的特征图之间进行有效的信息融合。

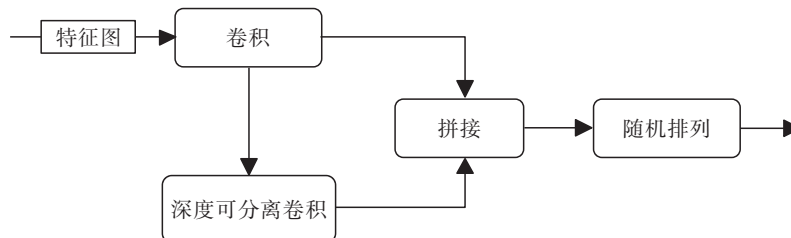


图4 GSConv模块结构图

通过图4可以清晰地看到GSConv模块的结构。首先,输入特征图。特征图先通过一个卷积层,随后通过深度可分离卷积层。深度可分离卷积层对输入特征图的每个通道独立进行卷积,将卷积层和深度可分离

卷积层的结果拼接起来。最后,拼接后的特征图经过随机排列操作(shuffle),重新排列特征通道,提高特征间的信息流动。

文章深入探讨了 GSConv 模块的应用潜力,并在此基础上进行了创新性的设计,重新打造了 GS Bottleneck 模块和 VOV-GSCSP 模块。其中,重新设计的 GS Bottleneck 模块如图 5 所示。这一模块不仅汲取了原始 Bottleneck 模块的精髓,更是在结构上进行了实质性的突破。在原始 Bottleneck 模块的基础上将传统的卷积模块替换为高效的 GSConv 模块。这一变革性的替换不仅极大地提升了计算的效率,使得模型在处理大规模数据时更游刃有余,而且在特征学习和表示能力上取得了显著的进步。GSConv 模块的引入使得模型能够更高效地捕捉和表示复杂的特征模式,从而提高模型的整体性能。此外,文章还将创新的模块设计思路应用于 VOV-GSCSP 模块中,进一步拓展了其在深度学习领域的应用范围。基于 GSConv 模块的重新设计,不仅提升了模型的计算效率,更在特征学习和表示上展现出了卓越的性能,为深度学习领域的发展注入了新的活力。

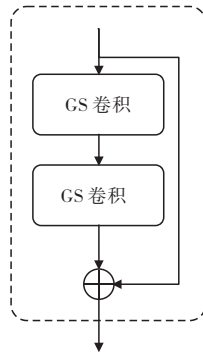


图5 重新设计的 GS Bottleneck 模块

为了重新设计 VOV-GSCSP 模块,本研究借鉴 C2f 模块,在 VOV-GSCSP 模块中引入了 split 操作,将输入特征图按照通道拆分成两个子特征图。这一举措不仅可以减少计算量和参数数量,提升计算效率,还能降低显存占用。同时,通过增加 GS 瓶颈模块的数量,VOV-GSCSP 模块能够有效地实现不同阶段特征图之间的信息融合。增加 GS 瓶颈模块数量的目的在于使每个 GS 瓶颈模块均能够从不同角度抽取输入特征,并将其压缩至更低维度的表示形式。通过多个 GS 瓶颈模块的叠加,逐渐对特征进行更深层次的抽象和压缩,生成更紧凑的特征表示。利用多个 Bottleneck 模块有效降低整体模型的参数数量和计算负荷,同时确保优良

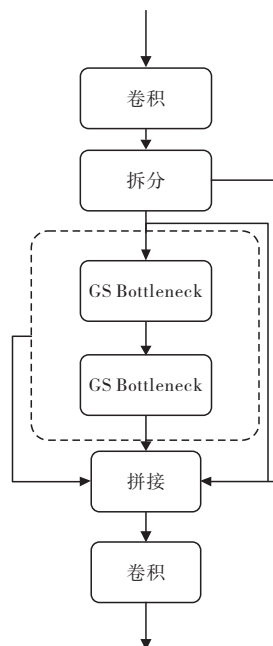


图6 重新设计的 VOV-GSCSP 模块

性能表现。本研究重新设计的VOV-GSCSP模块结构如图6所示。文章通过对VOV-GSCSP模块进行重新设计和优化,成功融合了C2f模块的概念,并引入了split操作,使得模块具备更高效的特征学习和信息融合能力,进一步提升了模型的性能和效率。

YOLO检测器被用于检测提取的所有背景掩码中的车辆,同时YOLO检测器还对提取的视频帧进行检测。首先对原始视频数据集进行检测,再对提取的背景进行检测,将检测结果保存为文档形式,并对检测结果进行后处理,详细标注每个检测结果是第几个视频的第几帧的检测结果,将其调整为可以输入异常检测器的格式,所有视频数据集都检测完成之后,会将提取的背景、生成的道路掩码以及目标检测结果输入异常检测器中,最终得到异常检测结果以及候选异常列表。

2.4 异常检测器

2.4.1 异常检测

在异常检测这一阶段,会获取到部分异常,包括异常的开始时间以及结束时间,而将可能出现异常的视频所对应的视频序号保存在候选列表中。可能出现异常的视频包括两种情况,分别是目标检测器在背景帧中没有检测到目标的视频以及在前景和背景之间没有匹配检测的视频,将其对应的视频序号写入候选异常列表之后,可以进行更加细粒度的异常检测。

文章使用比较背景与原始视频帧中的检测来识别异常情况。具体而言,对原始视频帧的检测框与背景的检测框进行比较,并计算它们之间的交并比(*IOU*),以确定它们之间的匹配程度,计算公式如式(1)所示。如果前景中的检测与背景匹配,则会增加前景中相应帧的帧得分。为了减少不必要的计算量,在计算交并比之前,文章首先使用外接矩形快速相交检测方法来判断两个检测框是否相交。

$$IOU = \frac{\text{inter_area}}{\text{box1_area} + \text{box2_area} - \text{inter_area}} \quad (1)$$

式(1)表示交并比的计算,其中box1_area表示检测框1的面积,box2_area表示检测框2的面积,inter_area表示检测框1和检测框2相交地方的面积。

在异常检测器中,文章设置了一个帧分数数组变量,将其初始化为0.在某一帧视频帧在与背景帧进行交并比计算之后会返回*IOU*的值,该*IOU*的值作为这一帧的帧分数,一帧视频帧中可能会出现多个车辆与背景帧中的车辆匹配,将*IOU*值进行累计,将每一帧的序号及其对应的帧分数写入帧分数文档中。为了获取异常事件的开始时间及结束时间,将帧分数文档传递到异常筛选器。首先读取帧分数文档,对于帧分数进行归一化处理,将每个分数除以最大分数,然后将第一个得分为1的帧分数对应的帧序号作为异常的开始,最后一个得分为1的帧分数作为异常的结束,将帧序号转变为时间作为异常的开始时间以及结束时间写入结果文档。

关于外接矩形快速判断方法,分别从水平方向和垂直方向判断。首先从水平方向投影重叠判断:检查两个检测框在水平方向上的投影是否有重叠。设第一个检测框的水平投影为 $[A, B]$,第二个检测框的水平投影为 $[C, D]$,则它们有重叠的条件是 $B > C$ 并且 $A < D$.如果条件成立,则表示两个检测框在水平方向上有重叠,否则没有重叠。从垂直方向投影重叠判断:同样,检查两个检测框在垂直方向上的投影是否有重叠。设第一个检测框的垂直投影为 $[E, F]$,第二个检测框的垂直投影为 $[G, H]$,则它们有重叠的条件是 $F > G$ 并且 $E < H$.如果条件成立,则表示两个检测框在垂直方向上有重叠,否则没有重叠。

2.4.2 细粒度异常检测

这一阶段主要致力于对异常列表中所对应的视频进行更加细粒度的检测。这些视频可能包含了异常区域。在某些情况下,异常区域小至 8×8 像素,由于小型车辆在普通视频帧中很难被检测到,因此在使用YOLO检测器之前,对视频帧进行一些预处理。具体而言,将视频帧分割成8个不重叠的块,这样形成了两行四列的分块结构。以一张 800×410 像素大小的图像为例,经过分块处理后,它将被划分成 200×205 个像素块,裁剪效果如图7所示。然后,分别对这些像素块进行检测,以提高检测精度和效率。对提取的背景视频帧也进行相同处理。通过这种方式,可以更全面地覆盖视频帧,更精细地进行异常检测。这种处理方法有助于增强检测器对小型异常区域的敏感度,提高异常检测的准确性和可靠性。

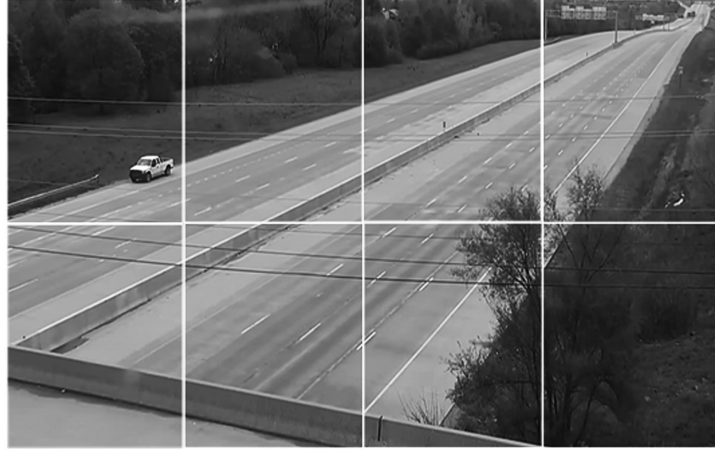


图7 裁剪效果

在本阶段,采用与上一小节相同的方法来获取异常情况。通过比较原始视频帧与背景是否匹配,检查原始帧中的检测结果是否与其对应的背景相匹配。最终,被确认的异常情况将作为异常的第二部分添加到异常文件中。通过这一过程,能够更加全面地识别异常情况,并对其标记准确的时间戳,以便后续的异常分析和处理。

3 实验

3.1 数据集

文章用的数据集是AI城市挑战赛赛道4数据集,这些视频由十字路口和高速公路沿线的多个摄像头提供。该数据集包含100个训练视频和150个测试视频,包括在各种恶劣天气条件下自然或异常发生的事件,如下雨、下雪或起雾,还包括晚上的视频数据集。每个视频的长度约为15分钟,录制速度为每秒30帧,分辨率为 800×410 像素。

3.2 评估指标

异常检测任务的评估将基于 F_1 分数来衡量模型的异常检测性能以及 $RMSE$ 来衡量检测时间的误差。具体而言, S_4 的分数计算公式如式(2)所示,其中 F_1 是 F_1 分数, $NRMSE$ 是归一化均方根误差($RMSE$)。 S_4 的分数在0~1之间,分数越高越好。

$$S_4 = F_1 \times (1 - NRMSE) \quad (2)$$

其中, F_1 分数是召回率和精确率的调和平均值。具体的,真阳性(TP)是指在真实异常的10秒内被正确检测到的异常事件数量。假阴性(FN)是指未能正确预测真实异常事件的数量。假阳性(FP)表示预测为异常但实际上不是真正异常的数量。 F_1 分数的计算公式如下

$$F_1 = \frac{2TP}{2TP + FN + FP} \quad (3)$$

归一化均方根误差($NRMSE$)表示所有真正预测的预测时间和实际异常时间的时间误差。为了消除提交过程中的抖动,将使用最小值为0、最大值为300的最小-最大归一化进行归一化处理,这表示任务的 $RMSE$ 值的合理范围。 $RMSE$ 大于300的团队将获得1的 $NRMSE$,因此 S_4 得分为0。 $NRMSE$ 的计算公式如下

$$NRMSE = \frac{\min\left(\sqrt{\frac{1}{TP} \sum_{i=1}^{TP} (t_i^p - t_i^g)^2}, 300\right)}{300} \quad (4)$$

其中, t_i^g 表示异常的实际开始时间, t_i^p 是由文章的异常检测方法得到的预测开始时间。

3.3 实验细节

本节的实验均在PyTorch神经网络框架下完成,PyTorch版本为1.13.1,Python版本为3.9,CUDA版本为12.1,使用的GPU为NVIDIA GeForce RTX 3060显卡。所用数据集包含150个视频,这些视频是AI城市挑战赛测试集的一部分。每个视频大约持续15分钟,除了第124个视频外,其他视频大约包含26000帧视频帧图片。第124个视频则有约52000帧视频帧图片。

在背景提取阶段,考虑到视频数据集中帧率多数为30,为了创建背景掩码,将帧率乘以60秒可得到1分钟的帧数。在处理视频帧时,检查当前处理的帧数能否被1分钟的帧数整除,若可以,则表示已经经过1分钟。然而,在生成的背景掩码中发现,有些视频只提取了8帧背景掩码,而其他视频提取了28帧背景掩码。通过观察视频数据集,发现一些帧率是30的倍数,还有些小于30,甚至还有帧率是浮点数。为解决这一问题,对这些不规则帧率进行取整操作,最终提取了正确的背景掩码。

3.4 实验结果

文章最终的实验结果如表1所示,其中conf是指目标检测中的置信度分数。可以看出,文章最终的实验结果相较于基准模型方法在异常检测精度方面提升了0.131%,其中 F_1 分数相较于基准模型方法提升了0.113%, $NRMSE$ 值降低了15.9639。

表1 异常检测结果

方法	F_1	$NRMSE$	S_4
baseline	0.7018	67.5044	0.5438
ours	0.7654	57.4251	0.6189
ours (conf > 0.5)	0.8148	51.5405	0.6748

对调整置信度分数conf提升异常检测精度分数情况进行如下分析。通过观察背景帧检测结果发现,视频的质量不高,存在误检的情况。因此,在对目标检测结果进行后处理时,只考虑检测结果置信度分数在0.5以上的检测框,消除了大量误检情况,进而提升异常检测的精度。具体地,用于检测框匹配的两个对象分别是视频背景帧中检测到的车辆以及视频帧中检测到的车辆。由于视频质量等原因,例如把一个目标错误地识别为车辆,将该目标的检测框与视频帧检测结果中的检测框进行匹配,那么就会将错误的异常检测结果写入异常事件文档,降低了 F_1 值,相对应的 S_4 分数也会降低,所以通过筛选掉低置信度分数conf的目标检测结果,能够提升异常检测精度。

结合异常检测结果的计算公式进行分析,异常检测结果受 F_1 分数和归一化均方根误差($NRMSE$)两个变量影响,其中 $NRMSE$ 值代表的是时间误差。通过实验结果可以看出, F_1 分数已经达到0.8148,而异常检测结果 S_4 分数却是0.6748,所以从降低时间误差方面入手,异常检测精度会进一步提升。



图8 视频帧

结合视频数据集进行进一步分析,得出以下结论:文章的视频数据集中,存在时间背景为夜晚的视频,即场景昏暗的视频,并且每辆车的车灯均保持亮起,这很不利于背景掩码的构建。通过观察使用高斯混合模型生成的背景,其视频帧如图8所示。可以看到右上角有很多车辆,这里出现了交通事故,但是由于视频质量、车辆目标距离监控较远以及车辆特征被灯光覆盖,目标检测器很难识别车辆,背景建模也无法获取效果很好的背景,所以很难检测到这种异常情况。

视频数据集还包括路况比较复杂的场景,场景中存在红绿灯的情况,这种情况很容易误检,也会导致精度下降。同时,采取每分钟提取一次背景的方法,会导致检测到的异常开始时间以及异常结束时间不够精

确。考虑每一帧都生成背景掩码或者每10秒构建一张背景掩码图片,这样异常检测器检测出的异常开始时间以及异常结束时间相较于之前会精确很多。

通过检查发现有异常的视频,有的视频存在卡顿情况,即几秒的视频全是一帧图片,这种情况也会导致误检,这属于视频数据集质量的问题,只能对异常检测结果进行检查,手动消除误检。

3.5 目标检测相关实验

3.5.1 数据集

在训练文章提出的检测模型时,共用了两个数据集,一个是VisDrone2019数据集,另一个是手动标注的数据集。具体地,首先提取AI城市挑战赛数据集中的训练视频帧,然后对这些视频帧进行筛选,筛选完之后进行标注,标注的训练集图片3120张,验证集图片780张。对于视频帧中的所有车辆均标注为汽车,同时对于VisDrone数据集也进行了预处理。由于VisDrone数据集采用XML作为标注文件,不能在YOLO检测器上进行训练,所以文章进行了预处理,将其标注文件转化为TXT格式。

3.5.2 实验细节

本节的所有实验均在PyTorch神经网络框架下完成,PyTorch版本为1.13.1,Python版本为3.8,CUDA版本为11.7,使用的GPU为NVIDIA GeForce RTX 4060 Ti显卡。在预训练方面,本节采用在COCO数据集上预训练的YOLOv8s模型作为所有模型的基线。模型参数设置如下:Epochs代表训练的迭代次数,设置为150;Batch_size代表每个迭代批处理的大小,设定为2,即每个批次处理2张图片;Imgsz表示送入网络模型中的图片尺寸大小,设置为1280像素。这些详细的参数设置和基准模型选择确保了实验的可重复性和可比性,同时提供了一个稳定的基础来评估方法的有效性。通过这些设置,研究人员能够在一致的环境下进行实验,更准确地评估模型性能,并为后续的实验设计提供参考和指导。

3.5.3 实验结果

本小节将深入分析本研究的实验结果。首先,重点关注在标注数据集上进行的消融实验,详细解读实验结果,探讨不同模型组件对检测性能的影响。其次,展示在VisDrone数据集上的实验结果进一步验证方法在实际场景中的有效性。最后,呈现本研究的定性结果,通过实例分析,直观展示模型在小目标检测任务上的优越性能。通过这些实验结果的综合分析,本研究将为轻量目标检测模型的优化与应用提供有力支持。

(1)在标注数据集上的消融实验:本研究针对提出的方法在标注数据集和VisDrone数据集上进行实验。在标注数据集上的实验结果如表2所示。由表2可明显看出,引入LightConv轻量卷积模块的有效性。相较于普通卷积,LightConv具有更少的参数和计算量,使得模型在相同计算资源下能够处理更大的感受野和更多的特征图,从而提高了模型的感知能力和表征能力,进而提升了精度和召回率。在引入VOV-GSCSP模块方面,实验结果显示与YOLOv8s模型相比,精确率、召回率和精度等指标均得到了提升。进一步加入GSConv卷积模块后,相较于YOLOv8s模型,各项评估指标也有所提升。尽管与仅添加VOV-GSCSP模块相比,除了精确率外其他指标都有所下降,但由于略微的精确率提升,保留GSConv模块。为提高召回率和精度,尤其是召回率,进一步添加LightConv模块。这些优化措施共同提升了模型的综合性能,使得在实验中取得更好的检测效果。文章所描述的实验结果清晰展示了不同模块的作用及相互之间的影响,为提高目标检测模型的性能提供了重要参考。

表2 在标注数据集上的实验结果

方法	Precision	Recall	AP50	AP50-95
YOLOv8s	0.880	0.861	0.921	0.572
+LightConv	0.879	0.880	0.935	0.595
+VOV-GSCSP	0.881	0.883	0.932	0.589
+VOV-GSCSP+GSConv	0.883	0.871	0.928	0.587
+VOV-GSCSP+GSConv+LightConv	0.881	0.888	0.937	0.598

(2)在 VisDrone 数据集上的实验结果:在 VisDrone 数据集上的训练中,迭代次数设置为 300 个 epoch,同时将(图像尺寸)设置为 640 像素。这里之所以选择将设置为 640 像素,是因为服务器显存无法支持整个训练过程,因此不得不降低图像尺寸以减少资源消耗。实验结果如表 3 所示。从表中的数据可以清楚地看出,改进模型不仅在自身数据集上取得了显著效果,而且在各项指标上都取得了显著提升。这进一步验证了检测模型在提升小目标检测精度方面的有效性。

表 3 在 VisDrone 数据集上的实验结果

方法	Precision	Recall	AP50	AP50-95
YOLOv8s	0.447	0.328	0.330	0.191
+VOV-GSCSP+LightConv+GSConv	0.607	0.459	0.491	0.300

3.5.4 定性实验结果

文章提出的模型在处理不清晰的视频帧时,表现出良好的适应性,对小目标的检测具有较强的鲁棒性,这种鲁棒性在异常停车检测任务中至关重要。在实际场景中,监控视频可能受到光照条件、天气或摄像头本身质量等因素的影响,导致视频帧的清晰度参差不齐。图 9 的定性实验结果验证了文章提出的检测模型在这些情况下依然能够准确地检测到小目标,进一步证明了文章提出的检测模型的有效性和实用性。



图 9 车辆检测网络模型定性实验结果

4 结语

文章旨在探讨轻量目标检测模型的优化与应用,特别是在小目标检测任务上的效能提升。为实现这一目标,文章基于先进的 YOLOv8 模型,创新性地引入了分组通道交换卷积模块,以期在减少模型复杂度的同时,提升其检测精度。为了验证新模型的有效性,文章选用了标注的数据集和 Visdrone2019 数据集进行实验。实验结果表明,新模型在小目标检测任务上表现出显著的优越性,证明了其在实际应用中的有效性。此外,文章还涉及道路掩码生成方法的添加、异常检测器的完善以及异常检测方法的优化等多个方面。这些改进工作的综合效果使得文章的异常检测方法具有更高的准确性和可靠性,为实际应用提供了更加坚实的基础。最终的异常检测结果表明,文章的方法在实际应用中具有良好的效果,能够准确识别并处理异常

情况。对实验数据的深入分析和评估进一步证实了文章所提出方法的有效性和可行性,与之前的方法相比,取得了明显的改进。

参考文献:

- [1] KRATZ L, NISHINO K. Anomaly Detection in Extremely Crowded Scenes Using Spatio-temporal Motion Pattern Models [C]. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, 1446–1453.
- [2] WANG T, SNOUSSI H. Detection of Abnormal Visual Events via Global Optical Flow Orientation Histogram [J]. IEEE Transactions on Information Forensics and Security, 2014, 9(06): 988–998.
- [3] CUI X, LIU Q, GAO M, et al. Abnormal Detection Using Interaction Energy Potentials [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011: 3161–3167.
- [4] COLQUE R V H M, CAETANO C, DE ANDRADE M T L, et al. Histograms of Optical Flow Orientation and Magnitude and Entropy to Detect Anomalous Events in Videos [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 27(03): 673–682.
- [5] FU Z, HU W, TAN T. Similarity based Vehicle Trajectory Clustering and Anomaly Detection [C]. IEEE International Conference on Image Processing, 2005, 2: 11–602.
- [6] CONG Y, YUAN J, LIU J. Sparse Reconstruction Cost for Abnormal Event Detection [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, 3449–3456.
- [7] LI A, MIAO Z, CEN Y, et al. Abnormal Event Detection based on Sparse Reconstruction in Crowded Scenes [C]. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, 2016, 1786–1790.
- [8] ZHAO B, FEI-FEI L, XING E P. Online Detection of Unusual Events in Videos via Dynamic Sparse Coding [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, 3313–3320.
- [9] LU C, SHI J, JIA J. Abnormal Event Detection at 150 Fps in Matlab [C]. Proceedings of the IEEE International Conference on Computer Vision, 2013, 2720–2727.
- [10] EDISON A, JIJI C V. Optical Acceleration for Motion Description in Videos [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, 39–47.
- [11] HERWIG J, LEBMANN S, BÜRGER F, et al. Adaptive Anomaly Detection Within Near-regular Milling Textures [C]. Proceedings of the IEEE International Symposium on Image and Signal Processing and Analysis, 2013, 113–118.
- [12] SHINE L, EDISON A, JIJI C V. A Comparative Study of Faster R-CNN Models for Anomaly Detection in 2019 Ai City Challenge [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019, 306–314.
- [13] BIRADAR K M, GUPTA A, MANDAL M, et al. Challenges in Time-stamp Aware Anomaly Detection in Traffic Videos [J]. arXiv preprint arXiv: 1906.04574, 2019.
- [14] BAI S, HE Z, LEI Y, et al. Traffic Anomaly Detection via Perspective Map based on Spatial-temporal Information Matrix [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019, 117–124.
- [15] SHINE L, CV J. Fractional Data Distillation Model for Anomaly Detection in Traffic Videos [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2020, 606–607.
- [16] DOSHI K, YILMAZ Y. Fast Unsupervised Anomaly Detection in Traffic Videos [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2020, 624–625.
- [17] DOSHI K, YILMAZ Y. An Efficient Approach for Anomaly Detection in Traffic Videos [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, 4236–4244.
- [18] ABOAH A. A Vision-based System for Traffic Anomaly Detection Using Deep Learning and Decision Trees [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, 4207–4212.
- [19] LI Y, WU J, BAI X, et al. Multi-granularity Tracking with Modularized Components for Unsupervised Vehicles Anomaly Detection [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2020, 586–587.
- [20] LI H, LI J, WEI H, et al. Slim-neck by Gsconv: A Better Design Paradigm of Detector Architectures for Autonomous Vehicles [J]. arXiv preprint arXiv, 2206, 02424, 2022.

Research on Anomaly Parking Detection Method based on Detection Box Matching

LI Ying-ying¹, LI Hai-fang^{1*}, SONG Rui-xia², LIU Zhan-dong¹, LI Ke¹, Ding Nan¹

(1. College of Computer Science and Technology, Xinjiang Normal University, Urumqi, Xinjiang, 830054, China;

2. Library, Xinjiang Normal University, Urumqi, Xinjiang, 830017, China)

Abstract: This article proposes a vehicle detection network model aiming at better detecting small vehicle targets, improving the accuracy of object detection, and achieving more robust anomaly detection. At the same time, the paper also explores an anomaly parking detection method based on detection box matching to improve the accuracy of anomaly detection. In order to optimize anomaly detection methods, a vehicle detection network model was introduced in the study to detect vehicle targets, and road masks were generated by combining target detection and target tracking methods. Through the comprehensive application of these methods, the anomaly detector has been further improved, achieving accurate acquisition of the start time and end time of abnormal events.

Keywords: Computer vision; Object detection; Abnormal detection; Background modeling; Road mask

(上接第7页)

Integrated EVE-NG, GNS3 and HCL to Build a Multi-vendor Network Simulation Experiment Platform

SUN Guang-yi

(Office of Network Security and Information Technology, Tianjin Conservatory of Music,

Tianjin, 300171, China)

Abstract: In response to the issue that EVE-NG and GNS3 simulation platforms mainly simulate Cisco devices, while HCL simulation platforms can only simulate H3C devices, and actual engineering projects often cover multiple vendor devices, this paper explores the integration of EVE-NG, GNS3, HCL and other network simulators with local physical network cards to build a network simulation experimental platform that integrates multiple vendor devices and can be used for actual network engineering projects. This article not only provides the basic principles of bridging and integrating the three network simulators mentioned above, but also outlines the specific bridging process with the local physical network card. After testing, it was confirmed that the integrated network simulation platform is compatible with simulating the operation of multi-vendor network devices.

Keywords: EVE-NG; GNS3; HCL; Pemu; Dynamips