

文章编号: 1005-0523(2026)03-0053-08



低等级路面病害的轻量化边缘实时检测方法

王靖智¹, 朱 路¹, 肖 乾², 黄德昌¹, 易钰程¹, 李 霄³, 熊 奎³

(1. 华东交通大学信息与软件工程学院, 江西 南昌 330013; 2. 华东交通大学载运工具与装备教育部重点实验室, 江西 南昌 330013; 3. 南昌市微轱联信息技术有限公司, 江西 南昌 330038)

摘要:针对传统路面检测车巡检效率低、设备成本高、实时性差等问题,提出一种基于边缘智能的轻量化路面病害实时检测方法。该方法采用 YOLO-Trip 模型,高效提取颜色和空间位置特征,结合 TensorRT 技术实现边缘端实时检测;针对现有里程测量存在的精度不足与部署困难等问题,设计了 IMU 和 GNSS 自校准高频里程计,结合卡尔曼滤波与线性插值算法,实现高频里程测量;构建了低功耗车载边缘计算平台,无需额外供电即可实时采集和检测路面图像。里程计对比实验表明,在 0~40 km/h 速度范围内,系统采样最大误差与车轮编码器相比仅为 0.9%,显著优于单 GNSS 方案;模型对比实验表明,YOLO-Trip 模型在召回率和平均精度上较基准模型均有所提升,同时参数量更少,计算量更小,减轻了边缘计算压力。系统能实时检测横向裂缝、纵向裂缝、网状裂缝和凹坑等病害,准确记录位置信息,适用于农村水泥路和山区柏油路,为路面养护提供数据支持。

关键词:边缘智能;目标检测;卡尔曼滤波;路面病害;轻量化

中图分类号:U418.4

文献标志码:A

本文引用格式:王靖智,朱路,肖乾,等.低等级路面病害的轻量化边缘实时检测方法[J].华东交通大学学报,2026,43(3):53-60.

Lightweight Edge Real-Time Detection Method for Low-Grade Pavement Diseases

Wang Jingzhi¹, Zhu Lu¹, Xiao Qian², Huang Dechang¹, Yi Yucheng¹, Li Xiao³, Xiong Kui³

(1. School of Information and Software Engineering, East China Jiaotong University, Nanchang 330013, China;
2. Key Laboratory of Transport Tools and Equipment of the Ministry of Education, East China Jiaotong University, Nanchang 330013, China; 3. VKELINE Information Technology Co., Ltd., Nanchang 330038, China)

Abstract: Aiming at the problems of low efficiency, high cost and poor real-time performance of traditional pavement detection vehicles, a lightweight edge real-time detection method for low-grade pavement distresses is proposed. This method adopts a novel YOLO-Trip model to efficiently extract color and spatial features, and integrates TensorRT technology to achieve real-time detection on the edge. For the existing mileage measurement challenge, an IMU and GNSS self-calibrated high-frequency odometer is designed, combined with Kalman filter and linear interpolation algorithm to realize ultra-high-frequency mileage measurement. A low-power onboard edge computing platform is built to collect and detect road surface images in real time without additional power supply. In the mileage measurement comparison experiment, the maximum sampling error of the system is only 0.9% different from that of the wheel encoder in the speed range of 0~40 km/h, which is significantly better than the single GNSS scheme. The model comparison experiment shows that the YOLO-Trip model leads the bench-

收稿日期: 2025-03-03

基金项目: 江西省研究生创新专项资金项目(2023-S478);江西省交通运输厅科技项目(2024QN008)

mark model by in recall rate and average precision, while the parameter quantity and the computational load are reduced, which alleviates the edge computing pressure. The system can detect transverse cracks, longitudinal cracks, alligator cracks and potholes and other diseases in real time, and accurately record the location information, which is suitable for rural concrete roads and mountainous asphalt roads, providing data support for road maintenance.

Key words: edge intelligence; target detection; Kalman filter; pavement distress; lightweight

Citation format: WANG J Z, ZHU L, XIAO Q, et al. Lightweight edge real-time detection method for low-grade pavement diseases[J]. Journal of East China Jiaotong University, 2026, 43(3): 53–60.

在高等级公路(如高速公路、国道)中,路面检测多依赖高性能大型检测车与专业人员标定,虽精度较高,但因设备体积大、操作要求高,难以实现低成本、高频次的巡检。相比之下,低等级公路目前仍主要依靠人工巡查或中型检测车:前者效率低下且主观性强;后者成本昂贵且检测频次有限,难以及时处置新发病害,对农村道路安全管理构成挑战^[1]。

因此,亟需一种轻量化、可搭载于普通民用车辆的实时路面检测方案。近年来,计算机视觉与深度学习的发展为该问题提供了新思路。已有研究尝试将深度学习用于路面病害识别,提升了检测效率与准确率,但多数缺乏真实场景部署验证^[2-4]。部分研究结合传统方法与深度模型,在实际环境中优化算法性能^[5-7],且对存储与带宽依赖大、检测效率低。为此,文献[8-9]引入边缘计算设备,将目标检测、图像分割等任务前置至终端,显著提升了检测效率并降低了网络带宽负载。

此外,路面检测系统还需兼顾硬件优化。当前里程测量多依赖车轮编码器,该方法虽检测精度较高,但存在安装复杂、设备体积较大等问题^[10]。针对农村道路巡检场景,亟需一种高精度且易部署的里程计量方案。文献[11]探索了基于惯性测量单元(IMU)的测量方法,在姿态估计、误差校正和三维校准等方面取得进展,以低成本实现以往需昂贵设备才能完成的功能,显著提升了实用性与经济性。

针对现有路面病害检测系统存在成本高、检测效率低、实时性不足等问题,本文利用边缘智能和信号处理等技术,研究低等级路面病害的实时检测方法。

1 基于边缘智能的系统架构

边缘智能将计算和推理部署在离数据源更接

近的位置,可以提供更快速、更安全、隐私保护更好的数据处理和决策能力,使得人工智能能够更好地应用于各种边缘计算设备和应用场景中。为了实现路面病害的实时、高效检测,本文提出基于边缘智能的低等级路面病害检测方案,该方案以低功耗高性能计算平台为核心,结合图像采集、里程计量、目标检测和数据存储模块,快速识别并分类路面的各种病害。这一过程不仅大大减少了对人力的依赖,同时也提高了检测的实时性和精度。基于边缘智能的系统架构还具有良好的可扩展性和灵活性,它可以通过云平台实现远程监控和管理,支持多点同步检测,从而为交通管理部门提供更加全面和及时的信息支持。

2 YOLO-Trip 目标检测算法

YOLO 模型(一种目标检测模型)在常规目标检测中取得了良好的效果,对于常规目标,其几何形态一般较为规整。然而,路面病害目标通常具有异构性,例如,沥青或水泥路面上的病害尺寸、形态等各不相同,导致基础 YOLO 模型检测精度不高,且推理速度较慢。

通过对路面病害样本的观察发现,虽然路面病害的尺寸、形态等存在较大差异,但就水泥路面病害而言,其颜色特征往往存在较大的相似性,即路面受损部分的颜色往往较深,多为棕黑色,与健康的水泥路面形成较大的颜色对比。此外,在空间分布方面,水泥路面与沥青路面的凹坑、网状裂缝等路面病害往往出现在行车道两侧,这是因为行车道车轮轨迹处长长期受车辆重压和频繁行驶的影响,导致路面磨损和破坏。根据低等级路面病害的这些特点,本文结合路面病害的颜色特征与空间位置特

征,通过在模型的特征提取层与特征融合层的关键位置引入 Triplet 注意力机制,提出 YOLO-Trip 模型,该模型能够更高效地学习路面病害样本的颜色、空间等特征。

在实现颜色注意力机制时,一种常见的方法是计算一个权重值,并使用该权重统一缩放特征图。为了得到这些通道的权重,该方法通过输入张量将全局平均池化降维到单一像素级别,导致大量空间信息丢失,从而破坏了通道维度和空间维度之间的相互依赖性。

为了解决上述不足,采用三重注意力(Triplet Attention)机制,它由三个并行分支组成,旨在融合跨维度的交互。Triplet Attention 第3个分支的输入张量经过 Z-Pool(MAX&AVG-Pool)操作后,形状从 $C \times H \times W$ 变为 $2 \times H \times W$,然后通过一个 $k \times k$ 的卷积核和 Sigmoid 激活函数生成空间注意力权重。这一过程类似于卷积块注意力模块(CBAM),都专注于在高度和宽度之间构建空间注意力。Triplet Attention 的前两个分支用来捕获输入张量的通道依赖性,即通过对三维矩阵进行转置操作的方式,在通道 C 与高度 H 、通道 C 与宽度 W 之间寻找依赖关系。最终,将这3个分支的结果通过取平均的方式进行聚合,以生成综合不同维度信息的注意力输出。此外, Triplet Attention 模块几乎不增加额外参数,在引入注意力机制的同时减小了卷积层中的通道压缩因子,有助于降低训练过程中所需的参数总量。这种设计不仅增强了模型对图像细节的关注能力,还保持了较高的计算效率^[12]。

3 高频里程计算法和电路设计

在路面病害检测作业中,相机图像采集的触发通常由检测车行驶的里程决定,车辆每前进至设定的距离,触发一次图像采集。传统路面检测车辆采用车轮编码器和全球导航卫星系统(GNSS)相结合的方式,实现距离测量和绝对位置采集。车轮编码器用于测量路段的长度信息,通过记录车轮的旋转角度和圈数,可以确定车辆的位置和行驶距离。然而,这种方式需对车辆机械结构进行改装,较为复杂,不适合轻量化作业设备,且轮胎胎压变化时轮胎周长也会改变,影响里程计量精度。

本系统采用 GNSS 结合 IMU 的方式替换车轮编码器。IMU 是一种用于测量加速度和角速度的

装置,可以精确测量车辆的姿态和运动状态。IMU 提供高精度和高频率的数据,可用于实时监测车辆的运动,并计算车辆的位置和行驶距离,而 GNSS 则提供车辆的绝对位置和速度信息,用于初始化和校准 IMU 的数据。通过融合 IMU 和 GNSS 的数据,减少 GNSS 系统定位精度受环境影响的问题,并提高里程计量的准确性和稳定性。相较于传统车轮编码器,此方案无需对车辆进行机械结构改装,且不受轮胎胎压不同而导致轮胎周长变化所带来的精度影响,仅需按角度固定于车顶即可实现高频率高精度的里程测量,在 40 km/h 行驶速度下,实现了 0.9% 的里程测量误差和 500 Hz 的数据刷新率。

由于 IMU 存在固有的积分漂移和传感器噪声问题。积分漂移是由于加速度计的累积误差导致位置和速度信息的偏移,传感器噪声会导致测量数据的不准确。为了减小这些误差和噪声对最终定位结果的影响,本文使用卡尔曼滤波对陀螺仪数据进行滤波。卡尔曼滤波是一种递归滤波算法,在测量数据和系统模型之间进行最优估计,可以有效提高数据的精度和稳定性。通过使用卡尔曼滤波,可以将 GNSS 提供的位置和速度信息与 IMU 提供的姿态信息相结合,得到更准确的运动状态估计^[13-14],这样可以提高车辆导航系统的精度和鲁棒性,使得系统在运动或者受到干扰时仍能获得较为可靠的定位和姿态信息。

3.1 高频 IMU 与 GNSS 融合算法

3.1.1 IMU 原始数据滤波

本研究实现了基于 MPU6050 传感器的运动姿态估计,包括加速度计和陀螺仪数据的读取、校正、滑动加权滤波以及卡尔曼滤波的处理流程,最终用于计算并输出修正后的加速度计数据 A_x 和 A_z ,分为4步。

第1步,加速度计数据处理与滑动加权滤波。原始加速度校正,表达式为

$$a_{cx} = a_x - \bar{a}_x, \quad a_{cy} = a_y - \bar{a}_y, \quad a_{cz} = a_z - \bar{a}_z + 1 \quad (1)$$

式中: a_{cx} , a_{cy} , a_{cz} 分别为经过校正的各方向加速度值; a_x , a_y , a_z 分别为未校正的加速度计读数,通过减去偏移量 $\bar{a}_x, \bar{a}_y, \bar{a}_z$,以消除静态偏差,偏移量为静止状态下一段时间内原始加速度值的均值。

原始角度计算,表达式为

$$\theta_x = \arctan\left(\frac{a_{cy}}{a_{cz}}\right) \times 57.295\ 78 \quad (2)$$

$$\theta_y = -\arctan\left(\frac{a_{cx}}{\sqrt{a_{cy}^2 + a_{cz}^2}}\right) \times 57.295\,78 \quad (3)$$

式中: θ_x 为横滚角; θ_y 为俯仰角; 57.295 78 为弧度与角度的转换系数。

滑动加权滤波, 表达式为

$$\theta_x^{\text{sum}} = \sum_{i=1}^{n_{\text{sample}}} i \cdot \theta_x^h[i], \quad \theta_y^{\text{sum}} = \sum_{i=1}^{n_{\text{sample}}} i \cdot \theta_y^h[i] \quad (4)$$

式中: θ_x^{sum} , θ_y^{sum} 分别为 x 和 y 方向的滑动加权和; n_{sample} 为采样队列长度; θ_x^h , θ_y^h 分别为 x 方向和 y 方向滑动窗口内的历史角度值序列; i 为当前窗口内样本的时间序号或位置索引; h 为“原始”观测值, 即滤波前的输入序列, 通过累加并乘以序列索引实现加权求和, 以平滑噪声。

最终角度计算, 表达式为

$$\theta_x = \frac{\theta_x^{\text{sum}} \times 0.211\,688}{n_{\text{sample}}}, \quad \theta_y = \frac{\theta_y^{\text{sum}} \times 0.211\,688}{n_{\text{sample}}} \quad (5)$$

式中: 0.211 688 为 MPU6050 (六轴运动处理传感器) 个体差异调整以及滑动加权滤波的综合系数, 每片 MPU6050 的系数值均不相同。

第 2 步, 陀螺仪数据处理与角速度积分。角速度计算, 表达式为

$$\Delta\omega_x = -\frac{\omega_x - \varpi_x}{131.0} \times dt, \quad \Delta\omega_y = -\frac{\omega_y - \varpi_y}{131.0} \times dt \quad (6)$$

式中: $\Delta\omega_x$, $\Delta\omega_y$ 分别为 x 方向和 y 方向经过校正和换算的角速度值; ω_x , ω_y 分别为 x 方向和 y 方向直接读取的陀螺仪角速度原始数据; ϖ_x , ϖ_y 分别为 x 方向和 y 方向静止状态下一段时间内原始角速度值的均值; dt 为单片机端程序运行两次的间隔时间; 131.0 为陀螺仪原始数据到角速度的换算系数。

积分运算, 表达式为

$$\Omega_x \leftarrow \Omega_x + \Delta\omega_x, \quad \Omega_y \leftarrow \Omega_y + \Delta\omega_y \quad (7)$$

式中: “ $\leftarrow$ ” 表示数据更新; Ω_x , Ω_y 分别为 x 方向和 y 方向角度变化的累加值, 使用累加形式代替积分运算。

第 3 步, 卡尔曼滤波。数据平均与方差计算, 表达式为

$$\begin{aligned} a_x[0] &\leftarrow a_x[1] & a_y[0] &\leftarrow a_y[1] \\ a_x[1] &\leftarrow a_x[2] & a_y[1] &\leftarrow a_y[2] \\ &\vdots & &\vdots \\ a_x[8] &\leftarrow a_x[9] & a_y[8] &\leftarrow a_y[9] \end{aligned} \quad (8)$$

$$S_x = \frac{1}{10} \sum_{i=1}^{10} a_x[i], \quad S_y = \frac{1}{10} \sum_{i=1}^{10} a_y[i] \quad (9)$$

$$R_x = \frac{1}{9} \sum_{i=1}^9 (a_x[i] - S_x)^2, \quad R_y = \frac{1}{9} \sum_{i=1}^9 (a_y[i] - S_y)^2 \quad (10)$$

式中: S_x , S_y 分别为 x 方向与 y 方向加速度的均值; $a_x[\cdot]$, $a_y[\cdot]$ 分别为 x 方向与 y 方向加速度协方差计算队列; R_x , R_y 分别为 x 方向与 y 方向加速度的方差。

卡尔曼增益与状态更新, 表达式为

$$K_x = \frac{P_x}{P_x + R_x}, \quad \Omega_x \leftarrow \Omega_x + K_x(\theta_x - \Omega_x), \quad (11)$$

$$P_x = (1 - K_x)P_x$$

$$K_y = \frac{P_y}{P_y + R_y}, \quad \Omega_y \leftarrow \Omega_y + K_y(\theta_y - \Omega_y), \quad (12)$$

$$P_y = (1 - K_y)P_y$$

式中: K_x , K_y 分别为 x 方向和 y 方向的卡尔曼增益; P_x , P_y 分别为 x 方向与 y 方向的预测误差协方差, 通过每次迭代更新。

第 4 步, 角度补偿与输出, 表达式为

$$A_x = a_x + \sin\left(\frac{\Omega_y}{57.295\,78}\right) \quad (13)$$

$$A_z = a_z - \cos\left(\frac{\Omega_y}{57.295\,78}\right) \times \cos\left(\frac{\Omega_x}{57.295\,78}\right) \quad (14)$$

式中: A_x , A_z 分别为经过重力分量消除后的 x 方向与 z 方向的加速度值。

3.1.2 线性插值

线性插值常用于融合 IMU 和 GNSS 的测量数据, 可以大幅提高定位位置的刷新率, 同时在一定程度上减小由于多径效应等影响带来的定位误差。该方法通过将两个 GNSS 测量点之间的时间分成 n 个间隔, 结合 IMU 测量的加速度和角速度数据, 逐一插值最终得到高刷新率的位置信息。

时间微分, 表达式为

$$\Delta t = t_1 - t_2 \quad (15)$$

式中: $t_1 - t_2$ 为程序两次运行的时间差值, 与上文 dt 含义相同但数值不相等。

计算额外位移 e , 表达式为

$$e = \frac{1}{2} \times A_x \times \Delta t^2 \quad (16)$$

速度更新, 表达式为

$$V \leftarrow V + A_x \times \Delta t \times 3.6 \quad (17)$$

式中: V 为在两次间隔为 100 ms 的 GNSS 数据中叠

加加速度数据的插值速度;3.6为将速度单位从m/s转换为km/h的转换系数。

位移更新,表达式为

$$D \leftarrow D + V \times 0.27778 \times \Delta t + e \quad (18)$$

式中: D 为最终结合匀速和加速运动的位移累加和;0.27778为将km/h转换为m/s的系数。

3.2 高频里程计电路设计

高频里程计的IMU模块和GNSS模块集成于同一电路之中。IMU模块采集MPU6050陀螺仪的实时数据,经卡尔曼滤波后叠加GNSS数据,随后通过USB总线传输至边缘计算设备,再由边缘计算设备经线性插值算法得到高刷新率的行驶里程。

4 计算平台与实验

4.1 模型训练环境配置

本文使用PyTorch平台进行路面病害检测模型的训练和推理,并借助图形处理器(GPU)加速训练和TensorRT加速推理,训练实验配置环境如表1所示。

表1 训练实验环境配置

Tab.1 Training experimental environment configuration	
名称	版本
操作系统	Ubuntu 22.04 LTS
编程语言	Python 3.10
显卡	NVIDIA GeForce RTX 2080 Ti
处理器	Intel Core i9-11900
内存	32 GB DDR4
硬盘	128 GB
深度学习框架	PyTorch 2.0.1

4.2 车载平台与配置

搭建了基于高速矩阵相机的测试平台,该平台还包括一辆五菱小型汽车、边缘计算设备、电源转换模块和高频里程计。在模型部署过程中,边缘计算设备采用了NVIDIA TensorRT技术和半精度量化技术以加速推理过程,使模型推理速度提升至原来的5倍,同时保持精度基本不变。TensorRT是由NVIDIA开发的用于加速深度学习推理的库。它通过优化网络结构、层融合、剪枝和量化等技术,可以大幅减少推理过程中的计算量和内存占用,从而提高推理速度。半精度量化是一种降低模型计算资源需求的技术。通过将模型中的浮点数参数和激活值转换为较低精度的表示格式(如16位浮点数),可以减少计

算和存储开销,从而加速推理过程。此外,TensorRT还支持动态张量处理和混合精度等高级优化技术,能够进一步压缩模型体积并提升计算效率。结合半精度量化,模型在保持较高精度的同时,能够显著降低功耗,这对于依赖非恒压稳定电源供电的边缘计算设备尤为重要。上述技术组合不仅适用于自动驾驶和智能交通领域,还可广泛应用于工业检测、医疗影像分析等需要高效实时处理的场景^[15]。车载边缘计算设备的详细配置与环境如表2所示。

表2 边缘计算环境配置

Tab.2 Edge computing environment configuration	
名称	版本
系统	Ubuntu 22.04 LTS
语言	Python 3.10
显卡	NVIDIA RTX A2000 Laptop GPU
处理器	Intel Core i3-10105F
内存	8 GB DDR4
硬盘	1 TB
推理框架	PyTorch 2.0.1
TensorRT	8.6.1

4.3 数据集

为评估本文提出的改进方法与量化方法,采用清洗后的GRDDC 2020数据集、RDD 2022数据集 中的中国区无人机拍摄部分(China Drone数据集)以及自采集数据集进行融合实验,总计23 991张样本。实验过程中,训练集、验证集、测试集的划分比例为8:1:1。后续实验均采用512×512像素的分辨率,部分数据样本如图1所示。

4.3.1 GRDDC 2020数据集

由于GRDDC 2020数据集未公布测试集标签,选用包含标签的21 041张训练集图片(分别来自捷克、印度和日本,数量依次为2 829张、7 706张和10 506张),共标注了25 046个边界框,涵盖9类路面病害。然而,该数据集中存在较多无关数据,故本实验选择经过清洗的数据集,仅保留包含纵向裂缝、横向裂缝、网状裂缝、凹坑4种类型的样本。清洗后,剩余图片20 085张。

4.3.2 China Drone数据集

China Drone数据集包含2 401张无人机拍摄的路面病害图片,共标注了3 068个边界框,涵盖4类路面病害,分别是:纵向裂缝(D00)、横向裂缝(D10)、网状裂缝(D20)、凹坑(D40)。



图1 数据集预览
Fig. 1 Dataset preview

4.3.3 自建数据集

自建数据集包含1 505张由后置高速相机拍摄的路面病害图片,共标注了4 507个边界框,涵盖4类路面病害,分别是:纵向裂缝(D00)、横向裂缝(D10)、网状裂缝(D20)、凹坑(D40)。

4.4 模型对比实验

为更直观地展示改进后YOLO-Trip模型的优势,采用以下评估指标:召回率(R)、精确率(P)、平均精度均值(mAP)、参数量(N)、计算量(GFLOPs,每秒十亿次浮点运算)、推理速度(F)以及训练时长(T),以全面评估模型性能。其中, P_1 为mAP@0.5, P_2 为mAP@0.5:0.95。

4.4.1 检测精度

如表3所示,改进的YOLO-Trip在精度指标上优于YOLOv5-S、YOLOv6-S、PP-YOLOE-S、YOLOv8-S和YOLOv10-S,展现出更高的检测精度。与RT-DETR-R18相比,YOLO-Trip在 P 和 P_1 上略低,但在 R 和 P_2 上更高,整体精度相当。

表3 精度对比

Tab.3 Comparison of accuracy

网络模型	P	R	P_1	P_2
YOLOv5-S	0.599	0.509	0.538	0.296
YOLOv6-S	0.607	0.514	0.546	0.300
PP-YOLOE-S	0.645	0.537	0.609	0.339
YOLOv8-S	0.681	0.567	0.637	0.360
YOLOv10-S	0.667	0.556	0.619	0.348
RT-DETR-18	0.688	0.587	0.655	0.364
YOLO-Trip	0.683	0.591	0.651	0.366

从表4可以看出,改进的YOLO-Trip在参数量和计算量方面均低于YOLOv6-S、YOLOv8-S和RT-DETR-S。此外,YOLO-Trip的训练时间也短于YOLOv6-S、PP-YOLOE-S、YOLOv8-S和RT-DETR-S,训练效率更高。结合表4的效率对比数据,YOLO-Trip在保持较低参数量和计算量的同时,实现了更高的训练效率。

表4 效率对比

Tab.4 Efficiency comparison

网络模型	$N/(\times 10^6)$	GFLOPs	T/h	$F/(\text{帧}/s)$
YOLOv5-S	7.02	15.8	8.763	11.7
YOLOv6-S	16.80	44.2	13.982	26.2
PP-YOLOE-S	7.90	17.4	8.890	12.1
YOLOv8-S	11.20	28.6	10.270	13.4
YOLOv10-S	7.20	21.6	9.200	12.9
RT-DETR-18	19.80	57.0	11.800	20.6
YOLO-Trip	9.79	26.0	9.679	13.7

4.5 车载设备实验

所有设备均安装在五菱宏光汽车上,用于在真实环境下测试路面病害检测能力。高速矩阵相机设置为最高分辨率1 280×1 024像素,并将曝光时间设置为自动模式,使相机在较暗环境中能够提高画面亮度,从而保持裂缝特征的清晰度。采集到图像后,首先将图像填充至1 280×1 280像素,随后压缩至512×512像素送入推理模型。该方法能够有效降低目标因图像尺寸变换带来的几何畸变,同时加速推理过程,以满足实时检测的需求。

4.5.1 高频里程计对比

为验证所提方案的有效性,选取GNSS与IMU融合方案、单GNSS方案以及编码器方案进行对比。编码器参数为1 000 P/R,数据采集平台和IMU里程计保持一致,以保证实验的公平性和测试环境的一致性。将三种方案部署在同一辆车上,同时进行数据采集。

选择一段长度约1.1 km的平坦路段进行测试,采样间隔设定为2.5 m,即车辆每行驶2.5 m采集一张路面图像。测试过程中,待车辆行驶速度提升至约40 km/h后开始测试。实验结果表明,单GNSS方案在采样间隔上的误差远大于IMU与GNSS融合方案,也远大于编码器方案,其采样间隔最大误差达44%。相比之下,编码器方案和IMU与GNSS融合方案的结果较为接近,采样间隔最大误差分别为0.5%和0.9%。

此外,针对行驶速度与采样间隔误差的关系进行了实验,采样间隔仍设定为2.5 m。结果表明,当车辆行驶速度越高时,采样间隔的误差也随之增大。单GNSS方案由于其采样速率仅为10 Hz,因此在行驶速度增加后,采样间隔的误差迅速增大。而IMU与GNSS融合方案和编码器方案由于具备超过500 Hz的采样频率,在车辆行驶速度不超过40 km/h时,均表现出较小的误差。

4.5.2 边缘计算设备推理测试

边缘计算设备采用的RTX A2000显卡的FP16算力为18.6 TFLOPS(每秒万亿次浮点运算),而作为对比的服务器端RTX 2080 Ti显卡(功耗300 W)的FP16算力为26.9 TFLOPS。前者的TDP(热设计功耗)仅为70 W,能够较好地胜任边缘智能应用场景。此外,通过TensorRT技术对模型进行加速优化,加速后的模型在推理时间上均有显著提升。

4.5.3 实际道路测试

在真实环境中进行了道路测试,路段长度为6.96 km,起点坐标为(28.779 678 3, 115.827 018 2),终点坐标为(28.772 396 7, 115.788 116 7)。其中农村低等级水泥路段和山区沥青路段约各占一半,部分验证了该系统对农村路面的适用性,也进一步证明该设备基于IMU和GNSS的方法对城市多建筑以及山区多丛林路段仍具有适用性。本次测试采用2.5 m间隔拍摄,单张图像纵向覆盖距离约为2.5 m,间隔采样精度为99.1%,即拍摄间隔控制在2.47~2.53 m。总计拍摄2 773张路面图像,其中存在路面

病害的图像共计497张。实验结果分析表明,横向裂缝在测试路段中出现的频次最高,共计435处;网状裂缝出现的频次最低,为22处。路面病害密集区域位于农村路段,这与人工清点及实际观察到的路面情况吻合,能够在一定程度上直观反映路面的质量状况。

5 结论

1) 提出了一种适用于低等级路面病害的轻量化边缘实时检测方法。通过边缘智能方式,实现路面病害的识别与定位。

2) 改进模型较基准模型平均精度提升1.4%,召回率提升2.4%。与同级别主流模型RT-DETR-R18相比,在检测精度大致相当的前提下,推理速度提升约50%。

3) 提出的高频里程计方法极大降低了系统部署难度,且在采样误差对比实验中,其精度与当前主流的里程测量方法相当。

4) 通过模型量化加速策略和无连接式里程测量方案,实现了低成本、易部署的边缘计算设备实时检测,提高了检测效率。该系统针对低等级路面,具有体积小、成本低、维护少等优点。

参考文献:

- [1] 易钰程,王靖智,朱路,等.路面缺陷智能检测系统与方法综述[J].华东交通大学学报,2023,40(5):19-31.
YI Y C, WANG J Z, ZHU L, et al. Review of the intelligent pavement defect detection system and methods[J]. Journal of East China Jiaotong University, 2023, 40(5): 19-31.
- [2] LIU Y Y, YE S T, WANG J Z, et al. Deep-supervised pavement crack segmentation algorithm for multiscale attention and feature fusion[J]. Photoelectron Laser, 2025, 36(11): 1140-1150.
- [3] YAO H, LIU Y H, LI X, et al. A detection method for pavement cracks combining object detection and attention mechanism[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(11): 22179-22189.
- [4] LIU F Y, LIU J, WANG L B. Asphalt pavement crack detection based on convolutional neural network and infrared thermography[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(11): 22145-22155.
- [5] 胡鹏,夏晓华,钟预全,等.采用多尺度特征增强的路面病害检测模型[J].西安交通大学学报,2025,59(2):156-169.

- HU P, XIA X H, ZHONG Y Q, et al. Pavement disease detection model based on multi-scale feature reinforcement[J]. Journal of Xi'an Jiaotong University, 2025, 59(2): 156-169.
- [6] 邓天民, 陈月田, 余洋, 等. 聚焦形状特征的路面病害检测算法[J]. 计算机工程与应用, 2024, 60(24): 291-305.
DENG T M, CHEN Y T, YU Y, et al. Pavement disease detection algorithm focusing on shape features[J]. Computer Engineering and Applications, 2024, 60(24): 291-305.
- [7] 高晶, 姚金杰, 刘鹏杰, 等. 基于改进YOLOv5s的路面病害检测[J]. 计算机系统应用, 2024, 33(9): 253-260.
GAO J, YAO J J, LIU P J, et al. Pavement disease detection based on improved YOLOv5s[J]. Computer Systems and Applications, 2024, 33(9): 253-260.
- [8] 白锋, 马庆禄, 赵敏. 面向航拍路面裂缝检测的AC-YOLO[J]. 计算机工程与应用, 2025, 61(1): 153-164.
BAI F, MA Q L, ZHAO M. AC-YOLO for aerial pavement crack detection[J]. Computer Engineering and Applications, 2025, 61(1): 153-164.
- [9] KAISER S, TOSUN A Ş, KORKMAZ T. Benchmarking container technologies on ARM-based edge devices[J]. IEEE Access, 2023, 11: 107331-107347.
- [10] WU Y R, GUO H F, CHAKRABORTY C, et al. Edge computing driven low-light image dynamic enhancement for object detection[J]. IEEE Transactions on Network Science and Engineering, 2022, 10(5): 3086-3098.
- [11] 王葱, 宋伟东, 孙尚宇, 等. 基于编码器和PLC的公路智能巡检测距模块设计[J]. 自动化仪表, 2023, 44(4): 12-17.
WANG C, SONG W D, SUN S Y, et al. Design of highway intelligent inspection distance measurement module based on encoder and PLC[J]. Process Automation Instrumentation, 2023, 44(4): 12-17.
- [12] LIU L, LI Z, LU R, et al. An enhanced smartphone GNSS/MEMS-IMU integration seamless positioning method in urban environments[J]. IEEE Sensors Journal, 2024, 24(24): 41251-41263.
- [13] WANG J Q, XIONG Z, ZHAO Y, et al. Covariance matrix transformation method for absolute/relative measurements fusion of vision/IMU/GNSS integration in parafoil landing[J]. IEEE Sensors Journal, 2024, 24(10): 16673-16687.
- [14] 陈博怀. 基于GNSS/IMU/UWB融合的扑翼机器人定位技术研究[D]. 南京: 东南大学, 2023.
CHEN B H. Research on positioning technology of flapping wing robots based on GNSS/IMU/UWB fusion[D]. Nanjing: Southeast University, 2023.
- [15] XIAO J P, CHEE J H, FEROSKHAN M. Real-time multi-drone detection and tracking for pursuit-evasion with parameter search[J]. IEEE Transactions on Intelligent Vehicles, 2024, 9(10): 6027-6037.



第一作者:王靖智(1999—),男,硕士研究生,研究方向为路面病害检测技术。E-mail: 1141359496@qq.com。



通信作者:朱路(1976—),男,教授,博士,研究方向为物联网,图像处理等。E-mail: lzhu@ecjtu.edu.cn。

(责任编辑:姜红贵)