文章编号：2096-1618(2023)06-0661-07

基于 ROS 与 YOLOv5s 的智能车障碍物检测导航系统的设计

李文海， 李超荣， 黄莹飞， 张 弛， 郭 伟
（马鞍山学院人工智能创新学院，安徽 马鞍山 243000）

摘要：针对智能车障碍物检测与自主导航任务中存在的识别准确率差、检出率低，以及自主导航路径规划器稳定性差等问题，设计一种基于 ROS 实验平台的障碍物检测与自主导航路径规划系统。系统以 YOLOv5s 作为障碍物检测算法框架，以 A* 算法与 TEB 算法融合作为自主导航算法框架，改善了智能车障碍物检测精度低与路径规划不稳定的问题。实验结果表明，搭载该系统的智能车能够完成障碍物检测、自主导航的任务，路径规划平均成功率达到 96.67%，障碍物检测准确率在 92% 以上，综合任务成功率在 90% 以上，具有障碍物检测准确率高，自主导航路径规划稳定性强的特性。

关 键 词：ROS；YOLOv5s；障碍物检测；路径规划；自主导航；智能车
中图分类号：TP249 **文献标志码**：A
doi：10.16836/j.cnki.jcuit.2023.06.007

0 引言

障碍物检测与自主导航路径规划是智能车的基本任务，也是该研究领域的一大热点。由于路面环境的复杂性与多变性，智能车行驶过程中捕捉的画面常出现光线不足或过强，障碍物遮挡等情况，使通过传统数字图像处理方法进行检测存在识别准确率差、检出率低等问题[1]。深度学习算法以强大的特征表征能力和泛化能力，可有效缓解识别准度差的问题。目前深度学习目标检测算法广泛采用的是 YOLO 系列算法，考虑到智能车存在的低算力限制，使用 YOLOv5 系列中网络规模较小的 YOLOv5s 算法框架[2]，具有推理速度快、检测精度高的优势，适用于有限计算能力的智能车障碍物检测的应用场景。

自主导航算法根据环境感知程度的不同，可分为全局路径规划算法与局部路径规划算法。全局路径规划算法是在已知的环境中进行路径规划，而局部路径规划算法则是在未知环境或部分可知环境的情况下进行路径规划。全局路径规划算法较为经典的是 A* 算法[3]、Dijkstra 算法[4]，由于算法结构简单、容易部署，常被广泛采用。局部路径规划算法更加侧重于当前的环境信息，对于效果较好的动态环境，常采用的是时间弹性带法（TEB）、动态窗口法（DWA）[5]。目前两种路径规划算法，多被单独采用，易出现路径规划失败或路径非最优解的情况。本文设计的系统融合全局路径规划 A* 算法与局部路径规划 TEB 算法，可同时进行路径规划，从而有效改善上述问题。

收稿日期：2022-11-12
基金项目：安徽省大学生创新创业训练计划资助项目（202213614003）

1 系统架构

智能车由软件层、驱动层、硬件层 3 部分构成。驱动层用于驱动底层硬件，软件层用于处理硬件层获取的数据。软件层包括 ROS 操作系统与 YOLOv5s 算法框架等，硬件层与驱动层主要包括各功能模组以及对应驱动，系统架构如图 1 所示。

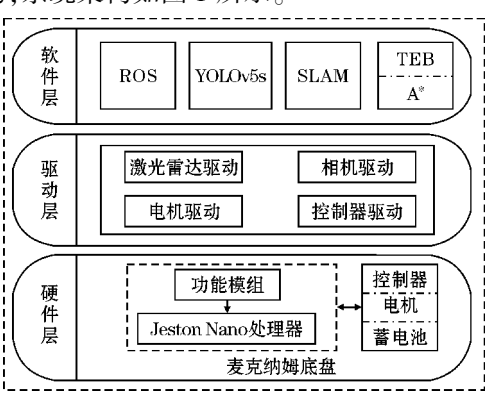


图 1 系统整体架构图

1.1 系统驱动与硬件架构

采用讯飞 U-CAR 晓 mini 版本智能车作为硬件平台，该智能车硬件架构如图 2 所示，主要分为上位机和下位机。上位机是智能车的中央控制系统，搭载 Jeston Nano 处理器和各功能模组，负责智能车环境数据的处理与分析。智能车功能模组主要涉及单目摄像头、激光雷达和 IMU 惯导模块。各功能模组需要搭载各自对应的驱动，以使单目摄像头获取图像数据，激光

雷达采集环境信息,IMU 惯导模块测量智能车在三维空间中的角速度和加速度数据。下位机由 STM32F4 控制器、蓄电池、电机和麦克纳姆轮构成。控制器驱动程序驱动 STM32F4,蓄电池为 STM32F4 控制器供电,电机驱动麦克纳姆轮。以 STM32F4 为核心组成的底层控制器,实现智能车的全方位移动。

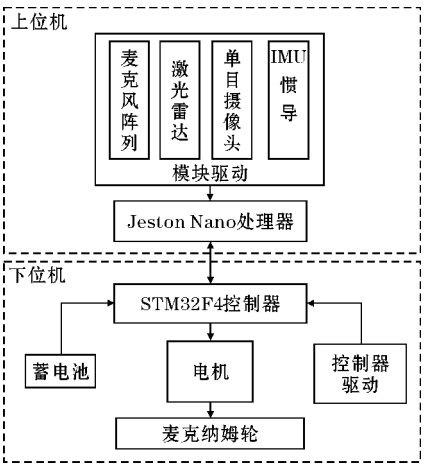


图 2 智能车硬件架构图

1.2 系统软件设计

系统软件组成如图 3 所示,其核心功能是完成障碍物检测和自主导航。障碍物检测功能经上位机的单目摄像头获取视频流,通过 YOLOv5s 模型实时输出障碍物检测信息。自主导航分为建图、路径规划和 PID(proportional integral derivative)控制 3 大模块。建图通过 SLAM 激光雷达,使用 Gmapping 算法[6]生成的 2D 地图,采用粒子滤波方式,将激光与姿势数据收集并创建栅格地图。路径规划采用全局路径规划和局部路径规划融合的方式,全局路径规划采用A* 算法进行构建,局部路径规划采用 TEB(timed elastic band)算法实现。TEB 算法在全局路径规划的基础上,满足各种约束条件,生成局部路径。经规划得到转向与转速等信息,转由下位机 STM32F4 控制电机的工作频率,进而改变麦克纳姆轮的速度与方向,实现 PID 控制。

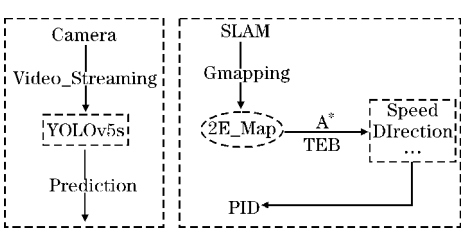


图 3 智能车软件组成

2 基于 ROS 的智能车路径规划算法

2.1 全局路径规划算法

全局路径规划是基于已知地图环境的静态路径规划算法,主要有A* 与 Dijkstra 两种算法[7]。Dijkstra 算法是一种广度优先搜索路径,优势在于得到路径多数较优,但缺陷在于每一个姿态点都需要遍历整个路径,效率低下,无法保证实时性。采用A* 算法作为全局路径规划器,与 Dijkstra 算法相比,优势在于引入启发路径代价函数,提高搜索效率。A* 算法的启发路径代价函数 $F(x)$,包括起始位置到达当前搜索点位置的代价函数 $G(x)$ 与当前搜索点位置到目标位置的代价函数 $H(x)$。A* 算法利用 $F(x)$ 作为判定下一搜索点位置的优劣,依次进行,直至到达终点。表达式如下:

$$F(x)=G(x)+H(x)$$

2.2 局部路径规划算法

局部路径规划算法是在全局路径的基础上,结合实时更新的代价地图,规划出合理的轨迹。常见的局部路径规划算法含有动态窗口法(DWA)[8]、模型预测控制法(MPC)[9]、时间弹性带法(timed elastic bands,TEB)等。

动态窗口法是在速度空间多次采样,通过评估函数不断对不同的速度与轨迹进行评价,并按照得分最高的速度与轨迹执行。该算法存在避障效果差、前瞻性不足等问题。模型预测控制法原理类似于 PID 控制器,以当前车辆与目标轨迹的差距作为评价函数,调整速度与前进方向。该算法存在计算量大,泛化能力弱等问题。时间弹性带法固定起始点与终点,可抽象为橡皮筋,皮筋上含有多个离散位姿约束,最终这些离散位姿组成的皮筋位置,即规划出的轨迹。TEB 模型优化了前两者的不足,有较强的前瞻性,避障效果好,但仍存在计算复杂度较大、速度和角度波动较大等问题。

2.2.1 TEB 算法原理

TEB 算法起源于 EB(elastic bands),EB 算法规划出的曲线可分为多段,其曲线抽象为橡皮筋,起始点与终点由全局规划器指定后,由于受到外力的影响以及上一小段皮筋的形变,下一段的橡皮筋也会发生形变。橡皮筋有外力的影响,自然就有收缩力与之达到力平衡,平衡后的曲线即最优路径。在 EB 算法的基础上,相邻的运动状态之间引入时间的概念,便于刻画运动状态,即为 TEB 算法[10],具体轨迹示例如图 4 所示。

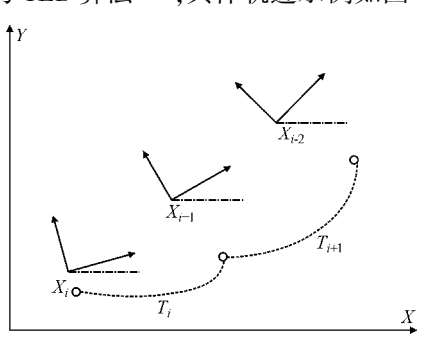


图 4 TEB 算法轨迹示例

2.2.2　轨迹约束条件

TEB 算法中的外力就是各个状态下的约束条件,包括跟随路径与避障约束、速度或加速度约束、运动学约束等。

(1)跟随路径与避障约束

跟随路径约束与避障约束相互制约,跟随路径施加外力将橡皮筋拉向全局路径,避障约束施加外力使得橡皮筋远离障碍物,二者受力方向恰好相反,图 5 所示为 x_i 到 x_{i+2} 的 3 种姿态关系。从 x_i 到 x_{i+1},跟随路径施加外力使得与全局路径差距逐渐缩小,但与此同时智能车不断靠近障碍物,距离也逐渐减小,容易碰撞到障碍物。所以,避障约束施加外力使之远离障碍物,x_{i+1} 到 x_{i+2} 的距离 d 逐渐增大。

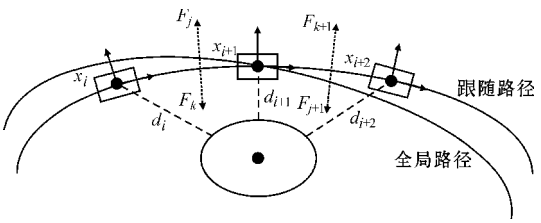


图 5　跟随路径与避障约束受力图

(2)速度或加速度约束

橡皮筋上的姿态点定义为一个向量$[x_i, y_i, \theta_i]^{\mathrm{T}}$,包含 3 个基本属性:坐标 x、y 和偏转角度 θ。智能车速度过快可能会因为麦克纳姆轮的抓地力不足,产生打滑现象。而速度过慢,则完成路径时间长,性能达不到要求。因此,本文对速度和加速度进行限制,即速度和加速度均要在最小值和最大值之间:

$$V_{\min} \leqslant F_v(B) \leqslant V_{\max}$$

$$a_{\min} \leqslant F_a(B) \leqslant a_{\max}$$

(3)运动学约束

智能车要在满足正确规划出路径且避开障碍物且不碰撞的前提下,尽量提高速度,行驶最优规划路径。

2.2.3　基于图优化的目标函数求解

图优化是通过节点和边来表示路径中各姿态之间的关系,节点为优化变量,边为优化变量之间的限制条件,目标函数求解可通过图优化转化为待优化顶点与待优化边的问题[11]。

智能车轨迹运动得到的目标函数是非线性的,非线性问题求解过程会消耗过多算力资源。因此须简化模型,将连续运动的曲线近似于无数多个离散的位姿,这样非线性问题就可转化为线性问题来求解。通过图优化来优化离散的位姿,使最终的离散位姿组成的轨迹能够达到时间最短、距离最短、远离障碍物等目标,同时满足所有的轨迹约束条件。具体 TEB 算法流程如图 6 所示。

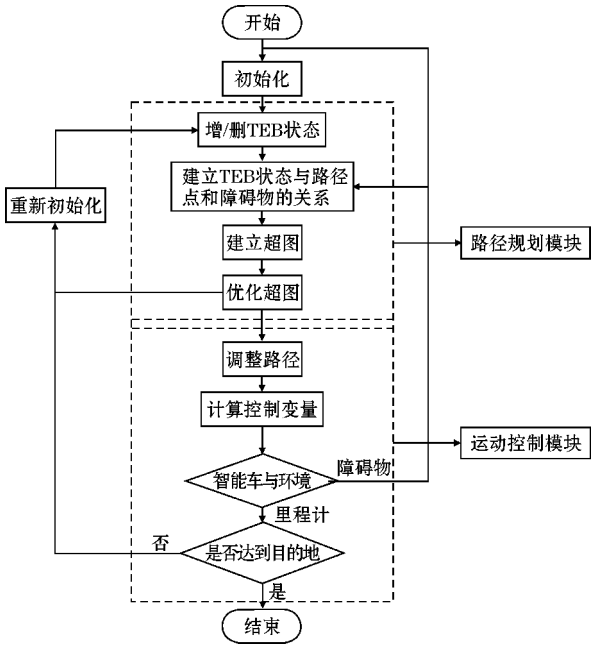


图 6　TEB 算法流程图

3　基于 YOLOv5s 的障碍物检测算法

3.1　数据集与预处理

实验数据集由智能车摄像头采集,共计图片 4600 张,分为 6 类,分别为人、自行车、汽车、摩托车、卡车、三角锥。为更好地表征学习,提升数据集的规模,对数据集进行不同程度的模糊处理、添加噪声、亮度变化、旋转等操作,如图 7 所示。将数据集扩充至 7000 张,按照 8∶1∶1 的比例划分为训练集 5600 张、验证集 700 张、测试集 700 张。

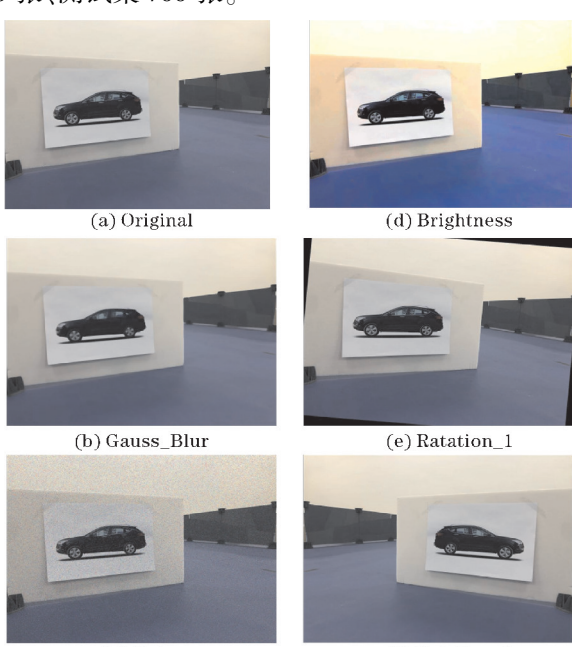


图 7　数据增强

3.2 YOLOv5s 网络模型

YOLOv5 根据不同的网络深度和网络宽度，可分为 YOLOv5s、YOLOv5m、YOLOv5l 以及 YOLOv5x[12]。考虑到智能车低算力的情况，采用网络规模较小的 YOLOv5s 模型进行障碍物检测，在保证高准确率的前提下仍可保持较快的识别速度。

YOLOv5s 网络结构如图 8 所示，该模型由输入网络(Input)、骨干网络(Backbone)、颈部网络(Neck)、检测网络(Prediction)构成[13]。输入网络中增加自适应锚框计算和图片缩放等操作；骨干网络和颈部网络采用不同的 CSP 结构，丰富梯度组合的同时减少计算量；检测网络中针对不同大小的目标，采用不同尺寸的特征图进行预测。

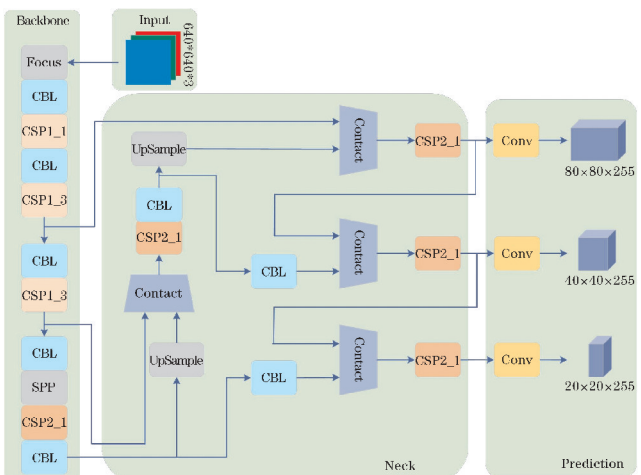


图 8 YOLOv5s 网络结构

3.2.1 输入网络

YOLOv5s 模型在输入网络中分别使用 Mosaic 数据增强、自适应锚框计算和自适应图片缩放 3 种操作。Mosaic 数据增强是由 CutMix 数据增强演变而来，CutMix 利用 2 张图片拼接，而 Mosaic 利用 4 张照片拼接，拼接的同时采用随机裁剪、随机缩放和随机分布的方式丰富数据集，使模型的鲁棒性提升。据先验知识可知，将数据集图片统一尺寸，检测效果会更佳。若直接对不同尺寸的图片，进行拉伸、收缩等操作，会造成图片失真。针对这一情况，输入网络中使用自适应图片缩放技术对图片进行缩放填充，即先将图片较长的边缩放至预设尺寸，短边通过灰色像素点填充。

3.2.2 骨干网络

骨干网络由 Focus、CSP 以及 SPP 模块构成，主要作用是对图片进行特征提取。Focus 模块主要作用是在空间关系没有被破坏的同时，增加感受野区域，由切片和卷积操作两部分组成。以 640×640×3 的一张图片为例，在每个通道间隔采样就得到 320×320×3 的特征矩阵，再经由 32 层卷积核卷积操作得到 320×320×32 的特征矩阵，即得到无信息丢失的二倍下采样特征图。YOLOv5s 模型设计了两种 CSP 结构，这种结构减少了网络参数数量，其中 CSP1_X 应用于骨干网络中，另外一种 CSP2_X 应用于颈部网络中。骨干网络中的 CSP 模块将浅层特征图一分为二，一部分通过 X 个残差单元(ResUnit)向后传播，与之直接卷积后的特征进行拼接，而颈部网络中的 CSP 模块将 X 个残差单元替换成 $2X$ 个 CBL 模块。SPP 模块目的是提取到更有益的特征信息，采用 3 种同一步长、不同卷积核尺寸的最大池化下采样，与不进行任何处理的特征图拼接，再通过卷积层实现降维。

3.2.3 颈部网络

颈部网络引用 PANet 的思想，采用 FPN(feature pyramid networks)与 PAN(perceptual adversarial network)结构，如图 9 所示。FPN 结构通过自顶向下进行上采样，使底层特征图包含更强特征信息。与之相反，PAN 结构是自底向上的，将高层特征通过下采样和低层特征融合。

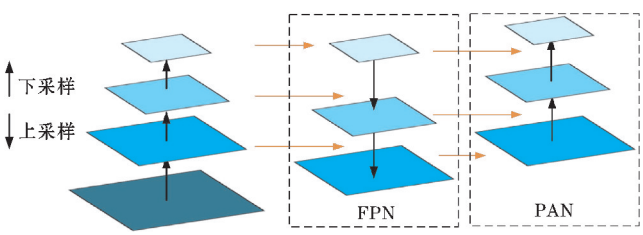


图 9 FPN 与 PAN 结构

3.2.4 检测网络

检测网络接收上层颈部网络输出的 3 种融合特征图后，各通过一次卷积操作，得到 3 种不同尺寸但维度相同的特征矩阵，用来检测不同尺寸的物体。其大小分别为 80×80×255、40×40×255、20×20×255，分别对小、中、大的物体进行检测。

4 超参数设置与评价指标

4.1 训练环境参数

为减少训练时间，GPU 采用 RTX 3080Ti，YOLOv5s 模型的训练环境配置如表 1 所示。

表 1 训练环境配置

类型	版本
CPU	12 核 Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz
GPU	RTX 3080 Ti
显存	11GB
Cuda	10.1
Cudnn	7.6.5
Tensorflow	2.3.0
Python	3.8.0

4.2　模型超参数

YOLOv5s 模型超参数设置如表 2 所示,模型迭代次数(iters)为 500,根据服务器性能指标,将批次(batchsize)设置为 32,学习率(learning_rate)作为影响网络的关键因素,将其设置为0.001。学习率值过大容易造成模型不易收敛,从而忽略最优值;而过小容易造成模型收敛缓慢,达到最大迭代次数时,仍没有找到最优解。权值衰减系数(weight_attenuation_coefficient)设置为0.0005,置信度(confidence)设置为0.6,即过滤置信度低于0.6的候选框。

表 2　超参数设置

参数名	数值
iters	500
batchsize	32
momentum factor	0.9
learning_rate	0.001
weight_attenuation_coefficient	0.0005
confidence	0.6

4.3　路径规划参数

智能车的路径规划参数对于能否规划出正确的路径,行驶的稳定性尤为关键。调试过程中使用动态调参工具调参,在 RVIZ 中可看见仿真环境地图。ROS 中的 TEB 功能包有智能车速度运动参数文件teb_local_planner_params. yaml、损失参数文件 costmap_common_params. yaml、全局代价地图参数文件 global_costmap_params. yaml、局部代价地图参数文件 local_costmap_params. yaml,其重要参数、变量值及其含义如表 3 ~ 表 6 所示。

表 3　teb_local_planner_params. yaml 参数

变量名	变量值	作用
acc_lim_x	0.72	最大 x 向加速度
acc_lim_theta	1.60	最大角加速度
min_turning_radius	0.15	最小转弯半径
max_vel_theta	1.6	最大转向角速度
free_goal_vel	False	消除目标速度限制
include_dynamic_obstacles	True	是否考虑动态障碍物

表 4　costmap_common_params. yaml 参数

变量名	变量值	作用
robot_radius	0.2	智能车的半径
obstacle_range	3.0	最大范围传感器读数
raytrace_range	3.0	清除指定范围外的空间

表 5　global_costmap_params. yaml 参数

变量名	变量值	作用
global_frame	map	全局代价地图中的全局坐标系
robot_base_frame	base_link	机器人的基坐标系
update_frequency	2.0	代价地图更新的频率
publish_frequency	0.5	发布信息的频率

表 6　local_costmap_params. yaml 参数

变量名	变量值	作用
global_frame	map	全局代价地图中的全局坐标系
robot_base_frame	base_link	机器人的基坐标系
update_frequency	10.0	代价地图更新的频率
publish_frequency	0.5	发布信息的频率

4.4　评估指标

YOLOv5s 模型采用精确率(Precision)、召回率(Recall)、调和值(F_1)平均精度(AP)和平均精度均值(mAP)作为评价指标。精确率是所有预测为正样本的结果中,预测正确的比率。召回率是所有正样本的结果中被正确预测的比率。调和值是精确率与召回率的调和平均数;平均精度是横坐标为召回率,纵坐标为精确率的二维曲线积分值。具体表达式如下:

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}}$$

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}}$$

$$F_1=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$$

$$\text{AP}=\int_0^1(\text{Precision})\,\text{d}(\text{Recall})$$

5　实验测试与结果分析

5.1　YOLOv5s 检测结果与分析

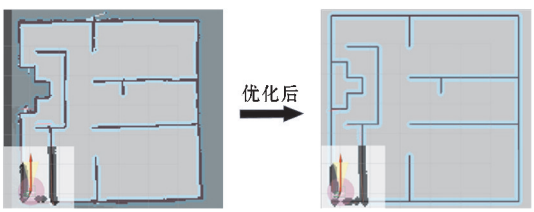
将服务器上训练好的 YOLOv5s 模型在测试集上进行测试,结果如表 7 所示。由表 7 可知,模型对人和三角锥的检测效果最佳,对与人这一类别的平均精度高达99.5%,三角锥类别平均精度为99.3%,其调和平均值F_1分别为97.7%和98.1%。这两类别效果好的原因在于,人在数据集中的占比较多,且三角锥颜色、形状等特征容易区分。摩托车类别在该模型上效果欠佳,平均精度仅达到92.6%。出现这种情况是由于摩托车在路面行驶速度较快,不易捕捉类别特征。综上,YOLOv5s 的模型检测结果符合实验预期,各类别平均精度均保持在 92% 以上,最高可达到 99% 以上,满足智能车行驶途中高精度的需求。

表 7　模型检测结果　　单位:%

类别	Precision	Recall	F_1	AP
人	99.1	96.4	97.7	99.5
自行车	95.6	94.4	95.0	96.7
汽车	97.8	99.1	98.4	98.9
摩托车	94.2	93.6	93.9	92.6
卡车	94.8	95.2	95.0	95.8
三角锥	98.9	97.4	98.1	99.3

5.2 联调测试结果与分析

本文采用 RVIZ 可视化呈现联调实验过程。首先导入栅格地图,如图 10(a)所示。该图使用 Gmapping 功能包建立完成,图中的智能车需不断移动直至扫描完全,受智能车抖动以及激光雷达精度影响,地图边界存在厚度不均匀现象。为保证实验严谨性与规范性,优化后的栅格地图如图 10(b)所示。之后,运行 YOLOv5s 模型对障碍物进行检测。最后,通过 RVIZ 中的 2D_positive 工具,发布坐标信息,使小车到达指定地点,完成自主导航任务。



(a)原栅格地图 (b)优化后的栅格地图

图 10 栅格地图

本次实验分别发布 3 个目标点进行路径规划测试并进行障碍物检测,实验场地具体放置如图 11 所示。鉴于场地限制,障碍物由障碍物图片代替,水平贴墙放置,图中蓝色粗线为障碍物放置点。智能车起点位于地图左下角,终点分别为 3 个终点位置。通过 RVIZ 分别发布 3 个目标点,进行联调测试。

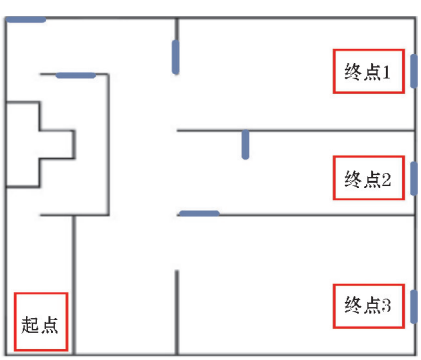
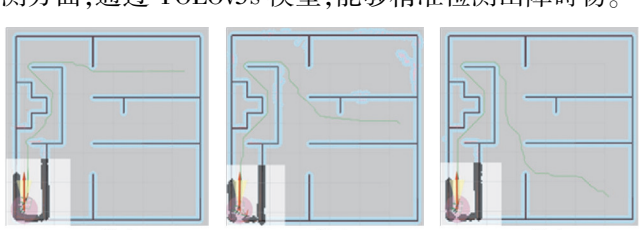


图 11 实验场地图

智能车路径规划情况与障碍物检测情况,分别如图 12、图 13 所示。智能车能够基于全局路径规划A*算法与局部路径规划 TEB 算法,完成起点与终点的路径规划,3 个不同目标点均能够正确规划出路线。障碍物检测方面,通过 YOLOv5s 模型,能够精准检测出障碍物。



(a) 终点1 (b) 终点2 (c) 终点3

图 12 路径规划情况

(a) 人 (b) 自行车 (c) 汽车

图 13 障碍物检测情况

本次实验记录了碰撞情况、障碍物误测情况、导航情况等。为避免偶然性的发生,实验分为 3 个终点,每个终点进行 30 次重复实验,共计 90 次实验。任务过程中无碰撞与误检,且成功路径规划,视为任务成功,实验结果如表 8 所示。智能车路径规划的导航平均成功率为96.67%,导航时长随着路线的长短依次增加,平均用时最快20.35 s,最慢27.53 s;障碍物检测方面,每组实验障碍物检出率均为 100%,最多误检 2 次,最低误检 1 次,在允许误差的范围内,能够达到高准度的要求;最终任务成功率分别为93.33%、90.00%、96.67%,均高达 90% 以上。搭载该系统的智能车能够完成障碍物检测、自主导航的任务,具有障碍物检测准确率高,泛化能力强,自主导航规划路径速度快、稳定性强的特点。实验联调测试结果表明,搭载 ROS 与 YOLOv5s 的智能车具有自主导航稳定性强,障碍物检测准确率高的特性。

表 8 实验结果

参数	终点位置		
	终点 1	终点 2	终点 3
实验次数/次	30	30	30
路径规划成功次数/次	29	28	30
任务成功次数/次	28	27	29
障碍物误检次数/次	1	2	1
平均导航耗时/s	20.35	23.88	27.53
障碍物检出率/%	100	100	100
路径规划成功率/%	96.67	93.33	100
任务成功率/%	93.33	90.00	96.67

6 结束语

设计了一种 ROS 与 YOLOv5s 的智能车动态障碍物检测导航系统。以 Jeston Nano、STM32F4、SLAM 激光雷达、单目摄像头、麦克纳姆轮等作为硬件基础,以 ROS 操作系统与 YOLOv5s 目标检测算法框架作为软件基础。实验结果表明,搭载该系统的智能车能够完成障碍物检测、自主导航的任务,路径规划平均成功率达到96.67%,障碍物检测准确率在 92% 以上,综合任务成功率在 90% 以上,具有障碍物检测准确率高,自主导航路径规划稳定性强的特性。

参考文献:

[1] 黄文涵. 基于单目视觉的智能车感知算法研究与应用[D]. 南京:东南大学,2021.

[2] 张凯祥,朱明. 基于 YOLOv5 的多任务自动驾驶环境感知算法[J]. 计算机系统应用,2022,31(9):226-232.

[3] 张强,鲁守银,张家瑞,等. 融合安全A*算法和改进人工势场法的巡检机器人路径规划[J]. 计算机时代,2022(11):29-33.

[4] 温淑慧,问泽藤,刘鑫,等. 基于 ROS 的移动机器人自主建图与路径规划[J]. 沈阳工业大学学报,2022,44(1):90-94.

[5] 郭烈,齐国栋,赵一兵,等. 融合A*与 TEB 算法的机器人多任务导航调度研究[J/OL]. 华中科技大学学报(自然科学版):2022-10-18.

[6] 常皓,杨巍. 基于全向移动模型的 Gmapping 算法[J]. 计量与测试技术,2016,43(10):1-4.

[7] 李涌. 基于激光 SLAM 的智能隧道巡检机器人自主移动平台研究[D]. 西安:长安大学,2021.

[8] 过佳颖. 基于 ROS 平台的导航机器人局部路径规划的研究与优化[J]. 现代信息科技,2022,6(5):144-148.

[9] 常宏. 基于 MPC 的局部路径规划与路径跟踪控制研究[J]. 汽车实用技术,2022,47(4):38-41.

[10] 贾屿. 基于 TEB 算法的无人驾驶汽车路径规划与避障技术研究[D]. 合肥:合肥工业大学,2021.

[11] 代婉玉,张丽娟,吴佳峰,等. 改进 TEB 算法的局部路径规划算法研究[J]. 计算机工程与应用,2022,58(8):283-288.

[12] 刘鹤,李建义. 基于 YOLOv5s 模型的车辆类型检测算法[J]. 廊坊师范学院学报(自然科学版),2022,22(3):24-28.

[13] 毛涛. 基于 YOLOv5 的小目标检测算法研究[D]. 淮南:安徽理工大学,2021.

Design of Intelligent Vehicle Obstacle Detection and Navigation System based on ROS and YOLOv5s

LI Wenhai, LI Chaorong, HUANG Yingfei, ZHANG Chi, GUO Wei

(Artificial Intelligence Innovation School, Ma'anshan University, Ma'anshan 243000, China)

Abstract: Aiming at the problems of poor recognition accuracy and low detection rate in obstacle detection and autonomous navigation tasks of intelligent vehicles, as well as poor stability of autonomous navigation path planner, this paper designed an obstacle detection and autonomous navigation path planning system based on ROS experimental platform. This system uses YOLOv5s as the obstacle detection algorithm framework and the fusion of A* algorithm and TEB algorithm as the autonomous navigation algorithm framework, which improves the accuracy of obstacle detection and makes path planning of intelligent vehicles more stable. The results show that the intelligent vehicle equipped with the system can complete the tasks of obstacle detection and autonomous navigation, and the average success rate of path planning is 96.67%, the accuracy rate of obstacle detection is more than 92%, and the success rate of comprehensive tasks is more than 90%. It has the characteristics of high accuracy rate of obstacle detection and strong stability of autonomous navigation path planning.

Keywords: ROS; YOLOv5s; obstacle detection; path planning; autonomous navigation; smart car