

分类号: TP391.4

密 级: 公开

学校代码: 10363

学 号: 2220320196



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题目 基于交通标志检测的无人车
路径规划研究

论文作者 王坤相

校内导师 王正刚（副教授）

校外导师 毛德平

专业学位类别 电子信息（085400）

研究方向（领域） 控制工程

论文提交日期: 2025 年 5 月 16 日

分类号: TP391.4

单位代码: 10363

密 级: 公 开

学 号: 2220320196

题 目 基于交通标志检测的无人车路径规划研究

题 目 Research on Unmanned Vehicle Path Planning Based
on Traffic Sign Detection

学生姓名: 王坤相

校内导师: 王正刚 (副教授)

校外导师: 毛德平

专 业: 电子信息

研究方向: 控制工程

论文答辩日期: 2025 年 5 月 12 日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：王坤祖

日期：2025年5月16日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在 _____ 年解密后适用本版权书。

本学位论文属于

不保密 ☒。

学位论文作者签名：王坤祖

指导教师签名：王坤祖

日期：2025 年 5 月 16 日

日期：2025 年 5 月 16 日

基于交通标志检测的无人车路径规划研究

摘 要

随着经济的迅速发展,无人驾驶车辆在日常生活中的应用日益普及,同时也引发了诸多新问题。其中,如何在严格遵守交通法规的前提下,为无人车规划出既合规又高效的行驶路线,对维护交通秩序和保障行车安全至关重要。为此,本文以无人驾驶车辆检测到的路标信息为约束条件,在确保严格执行交通法规的基础上,设计出最优化的行驶方案。本文的主要研究内容包括:

(1) 针对当前交通标志检测中小目标识别精度低、漏检、检测不准确的问题,提出一种基于 YOLOv8s 的高精度模型。该模型采用 RepGhost Bottleneck 代替了传统 C2f 中的 Bottleneck 模块,形成 C2fRepGhost 结构,降低模型计算量;在主干网络中融入 CBAM(Convolutional Block Attention Module)注意力模块,增强特征的语义和位置信息的综合表达能力,以强化对交通标志特征的捕捉;针对小目标识别问题,增设了专门的小目标检测层,以提升小目标的检测准确性;此外,通过使用 MPDIoU 损失函数替换标准损失函数,优化了收敛速率和回归准确度。在 TT100K 数据集上的实验显示,经优化的网络模型在准确率、召回率和 mAP(mean Average Precision)指标上分别获得了 4.8%、9.7%和 6.1%的提升,计算量下降了 0.5GFLOPS,证实了在几乎不增加网络复杂度的情况下,检测性能得到了显著提高。

(2) 提出了一种基于改进蚁群混合人工势场的路径规划算法。

该算法首先构建非均匀分布的初始信息素，引导蚂蚁快速向目标收敛，从而减少搜索时间并加速收敛过程；在搜索过程中，采用自适应信息素更新策略，根据路径质量动态调整信息素浓度，以增强全局搜索能力；同时，结合改进的人工势场法，利用引力与斥力调控机制优化避障策略；为避免算法陷入局部最优，设计了基于惩罚系数的动态信息素调整机制，以增加搜索随机性并提升跳出局部最优的能力；此外，采用三次 B 样条曲线对路径进一步平滑优化，使无人车行驶轨迹更加连续和自然。实验结果表明，在不同障碍物密度环境下，改进算法在路径长度上平均缩短了 3.995 米，运行时间降低了 15.8%，同时路径平滑性明显提升，充分证明了改进算法在路径规划方面的优势。

(3) 将本文所改进的算法部署到 HW-Autocar 无人车平台，在室内搭建场景，构建栅格地图，并融合交通标志与障碍物信息，设计左转与右转交通标志场景进行实验。实验结果表明，无人车能够精准识别交通标志，并基于检测结果规划合理路径，成功到达目标点，证明了本文所提方法的可实施性。

关键词：无人车，交通标志检测，YOLO，路径规划，人工势场，蚁群算法

RESEARCH ON UNMANNED VEHICLE PATH PLANNING BASED ON TRAFFIC SIGN DETECTION

ABSTRACT

With the rapid development of the economy, the application of driverless vehicles in daily life has become increasingly popular, and at the same time, it has also given rise to many new problems. Among them, how to plan both compliant and efficient driving routes for unmanned vehicles under the premise of strict compliance with traffic regulations is crucial for maintaining traffic order and ensuring driving safety. To this end, this paper takes the road sign information detected by the unmanned vehicle as the constraints, and designs an optimized driving scheme on the basis of ensuring the strict enforcement of traffic regulations. The main research content of this paper includes:

(1) Aiming at the current problems of low target recognition accuracy, leakage and inaccurate detection in small targets in traffic sign detection, a high-precision model based on YOLOv8s is proposed. The model adopts the RepGhost Bottleneck instead of the Bottleneck module in the traditional C2f to form the C2fRepGhost structure, which reduces the amount of model computation; incorporates the CBAM (Convolutional Block Attention Module) attention module into the backbone network,

which enhances the features' semantic and location information to enhance the comprehensive expression ability of features to strengthen the capture of traffic sign features; for the small target recognition problem, a dedicated small target detection layer is added to improve the detection accuracy of small targets; in addition, the convergence rate and regression accuracy are optimized by replacing the standard loss function with the MPDIoU loss function. Experiments on the TT100K dataset show that the optimized network model obtains 4.8%, 9.7%, and 6.1% improvements in accuracy, recall, and mAP (mean Average Precision) metrics, respectively, and the computational effort decreases by 0.5 GFLOPS, which confirms that the detection performance has been significantly Improvement.

(2) A path planning algorithm based on improved ant colony hybrid artificial potential field is proposed. The algorithm firstly constructs a non-uniformly distributed initial pheromone to guide the ants to converge to the target quickly, so as to reduce the search time and accelerate the convergence process; during the search process, an adaptive pheromone updating strategy is adopted to dynamically adjust the pheromone concentration according to the quality of the path, so as to enhance the global search ability; at the same time, combined with the improved artificial potential field method, the obstacle avoidance strategy is optimized by using the gravitational force and repulsive force regulating mechanism; in order to avoid the algorithm In order to avoid the algorithm

from falling into the local optimum, a dynamic pheromone adjustment mechanism based on the penalty coefficient is designed to increase the randomness of the search and enhance the ability of jumping out of the local optimum; in addition, the path is further smoothed and optimized by using the cubic B-spline curve, so as to make the driving trajectory of the unmanned vehicle more continuous and natural. The experimental results show that under different obstacle density environments, the improved algorithm shortens the path length by an average of 3.995 meters and reduces the running time by 15.8%, while the path smoothing is obviously improved, which fully proves the advantages of the improved algorithm in path planning.

(3) The improved algorithm in this paper is deployed to the HW-Autocar unmanned vehicle platform, the scene is built indoors, the raster map is constructed, and the traffic signs and obstacle information are fused, and the left-turn and right-turn traffic sign scenes are designed for experiments. The experimental results show that the unmanned vehicle can accurately recognize the traffic signs and plan a reasonable path based on the detection results to successfully reach the target point, which proves the implementability of the method proposed in this paper.

KEY WORDS: unmanned vehicle, traffic sign detection, YOLO, path planning, artificial potential field, ant colony algorithm

目录

摘 要.....	I
ABSTRACT.....	III
第 1 章 绪论.....	1
1.1 课题研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 无人车研究现状	2
1.2.2 目标检测算法研究现状.....	4
1.2.3 路径规划算法研究现状.....	7
1.3 论文主要研究内容与章节安排.....	8
第 2 章 相关背景知识介绍.....	10
2.1 卷积神经网络.....	10
2.1.1 输入层.....	11
2.1.2 卷积层.....	11
2.1.3 激活函数.....	12
2.1.4 池化层.....	13
2.1.5 全连接层.....	13
2.2 智能小车运动学模型.....	14
2.3 环境建模.....	16
2.3.1 环境地图介绍.....	16
2.3.2 地图栅格化.....	18
2.4 本章小结.....	19
第 3 章 基于改进 YOLOv8s 的交通标志检测	20
3.1 YOLOv8s 模型	20
3.2 改进的 YOLOv8s 模型	21
3.2.1 网络结构轻量化设计.....	22
3.2.2 引入 CBAM 注意力机制.....	24

3.2.3 针对小目标增加预测头.....	26
3.2.4 MPDIoU 损失函数.....	27
3.3 交通标志检测实验与结果分析.....	28
3.3.1 训练环境和数据集.....	28
3.3.2 实验性能评价指标.....	28
3.3.3 消融实验分析.....	29
3.3.4 对比试验分析.....	32
3.4 本章小结.....	33
第 4 章 基于改进蚁群混合 AAPF 的路径规划算法	35
4.1 人工势场法.....	35
4.1.1 传统人工势场法.....	35
4.1.2 改进的人工势场法.....	38
4.2 蚁群算法.....	39
4.2.1 蚁群算法相关参数.....	40
4.2.2 蚁群算法基本原理.....	41
4.3 改进的蚁群算法.....	43
4.3.1 初始化信息素分布.....	43
4.3.2 自适应信息素浓度.....	44
4.3.3 惩罚系数.....	45
4.4 混合算法实现流程.....	46
4.5 仿真实验分析.....	48
4.5.1 少量障碍物环境中仿真结果.....	48
4.5.2 中等数量障碍物环境中仿真结果.....	50
4.5.3 较多数量障碍物环境中仿真结果.....	51
4.5.4 B 样条曲线平滑	52
4.5.5 融合交通标志的环境仿真结果.....	53
4.6 本章小结.....	54
第 5 章 实验与分析.....	55
5.1 无人车实验平台搭建.....	55

5.1.1 无人车硬件结构.....	55
5.1.2 无人车软件环境.....	56
5.1.3 主从机通信机制.....	56
5.2 实验设计流程.....	57
5.2.1 环境地图构建.....	57
5.2.2 地图信息完善.....	58
5.2.3 无人车自主导航.....	59
5.3 实验结果分析.....	59
5.4 本章小结.....	63
第 6 章 总结与展望.....	64
6.1 主要工作总结.....	64
6.2 进一步工作展望.....	65
参考文献.....	66
攻读硕士学位期间主要研究成果.....	72
致 谢.....	73

第 1 章 绪论

1.1 课题研究背景及意义

汽车行业的迅猛发展在带来便捷交通和经济增长的同时，也引发了诸多问题。随着科技进步与经济发展，汽车保有量持续攀升，交通事故数量不断增加，这不仅造成了严重的人身财产损失，还使交通拥堵和交通安全问题备受关注^[1]。根据最新统计数据，在 2018-2024 年期间，我国共发生道路交通事故约 150 万起，其中 88% 的事故是由车辆违反交通规定引发，多车碰撞事故占比达 72%，因颅脑损伤导致致命后果的事故比例为 8.5%。这些数据深刻地反映出当前交通状况的严峻性，也凸显了寻求有效解决方案的紧迫性。在此背景下，无人驾驶技术作为一种极具潜力的创新手段，逐渐进入人们的视野，有望成为解决交通问题的关键途径。

无人驾驶技术是一个融合多学科的新兴领域，其包含环境建图、路径规划和运动控制等模块，其中路径规划尤为关键^[2]。然而，现有全局路径规划算法存在效率低、易陷入局部最优的状况，且对如何结合交通标志与障碍物信息规划路径的研究相对不足^[3]。无人车依靠视觉相机和激光雷达采集环境信息^[4]，视觉相机可获取交通标志图像，但现有检测技术在精度、准确性与实时性的平衡上仍有欠缺；激光雷达虽能扫描障碍物轮廓，却无法获取交通标志图像，且在强光或黑夜等特殊环境下，摄像头检测效果差，凸显出研究基于深度学习目标检测算法的重要性。

移动机器人技术的发展历程悠久，应用广泛，在智能制造、环境监测、医疗护理、智能交通等众多领域都发挥着重要作用，但目标检测和自主导航始终是该领域面临的核心难题，尽管这些技术取得了长足进步，但在实际应用中仍面临检测精度、准确性与实时性兼顾困难，路径规划不够优化以及在复杂环境中导航效率低等问题。

基于深度学习的目标检测和路径规划研究有助于推动无人驾驶系统精度和实时性的提升，进而增强无人驾驶汽车的安全性，为其在复杂交通环境中的应用

提供理论支持^[5]。深入探索无人车交通标志识别与路径规划问题，能够填补当前研究的空白，完善相关理论体系，推动汽车工业及相关领域技术的发展。

无人驾驶技术依托交通标志检测算法精准识别标志信息，并结合路径规划算法，实现符合交通规则自主行驶。然而，在复杂任务环境下，智能车的运行面临诸多挑战。因此，研究基于交通标志检测的无人车路径规划，不仅有助于缓解交通拥堵、降低碰撞事故发生率、减少人身财产损失，还能为构建智能交通系统提供精准数据支持，推动无人驾驶技术的进一步发展。

1.2 国内外研究现状

1.2.1 无人车研究现状

(1) 国外无人车研究现状

无人驾驶汽车的发展可追溯到 1939 年，在纽约世界博览会上，美国通用汽车第一次展示了电动无人驾驶汽车，首次向众人呈现了这一超前概念^[6]。此后，1953 年，美国贝瑞特电子公司研发出全球首台自主导航车，但其运行受限于固定轨道，应用范围较为单一。1995 年，美国卡耐基梅隆大学“Navlab-V”无人车成功完成东西部穿越挑战，推动了该领域技术的进一步探索^[7]。

2005 年，美国斯坦福大学改装的车辆在无人驾驶“大挑战”中成功完赛，克服了颠簸的山路与湿地复杂环境，引发了行业对无人驾驶技术更多的关注与研究热情^[8]。2011 年 6 月，美国内华达州立法批准无人驾驶汽车进行上路测试，谷歌获授权，使其成为首个赋予无人驾驶汽车合法资格的地区，这为行业发展提供了重要的法律支持与实践平台。次年 5 月，谷歌获得首个无人驾驶汽车牌照，开启了合法上路测试的新篇章，引领了行业发展潮流。2013 年，奔驰对 S500 进行了智能驾驶改装，在德国路面完成了 100 公里测试，实现城市内及城市间的无人驾驶行驶，展示了传统汽车制造商在该领域的技术实力与创新成果。2014 年 5 月，谷歌发布最新无人驾驶汽车原型，持续推动技术迭代升级^[9]。2015 年 10 月，特斯拉将 Autopilot 投入商用，成为首个实现自动驾驶技术商用化的企业，加速了无人驾驶技术在市场中的应用普及。

2016 年 12 月，谷歌成立 Waymo，专注于无人驾驶技术研发与商业拓展。到

2018 年初, Waymo 公司无人车在加州上路测试成绩斐然, 每 1000 英里人类干预仅 0.18 次, 累计测试超 400 万英里, 且自主设计的激光雷达和计算机视觉系统证明了技术的可靠性与成熟度。同年 2 月, 谷歌宣布计划生产数千辆无人驾驶出租车投入运营, 并推出 WaymoAPP, 开启了无人驾驶技术商业运营的新征程。

2019 年, 英伟达推出了新一代自动驾驶计算平台, 为无人车提供更强大的算力支持, 宝马等车企开始采用该平台进行无人车技术研发, 使得车辆对复杂路况的感知和决策速度大幅提升。2020 年, 疫情背景下, 无人配送车迎来发展机遇, 亚马逊的 Scout 无人配送车在美国多个社区进行配送服务试点, 它能够在人行道上自动行驶, 避开障碍物, 将包裹准确送达客户家门口, 有效减少了人员接触。2021 年, 通用汽车旗下的 Cruise 无人车在旧金山街头开展商业化运营试点, 为乘客提供无人驾驶出租车服务, 其车辆配备了先进的传感器和智能算法, 能够应对城市道路中的各种交通状况。2022 年, Waymo 与极星汽车合作, 推出基于极星 2 改造的无人车, 该车在激光雷达、摄像头等传感器的布局 and 性能上进行优化, 提升了对周围环境的感知精度。2023 年, 德国博世公司开发的一套全新无人驾驶系统, 应用于部分商用货车上, 实现了在高速公路上的编队自动驾驶。

(2) 国内无人车研究现状

我国无人驾驶汽车领域的起步相对滞后, 与国外存在一定技术差距。自 1980 起, 我国开启了对无人驾驶汽车的研究征程。1992 年, 国防科技大学成功研制出中国首辆无人驾驶汽车, 这标志着我国在这一领域取得了重大进展^[10]。2003 年, 又与一汽联合研制了红旗 CA7460 无人驾驶汽车, 该车可实现自动变道避障^[11]。2005 年, 上海交通大学也成功研制出无人驾驶汽车, 实现自主感知、路径规划与避障能力^[12]。2011 年, 国防科技大学再次与一汽联合研发无人车, 并且在高速公路上完成了 286 公里的测试, 刷新了无人驾驶在复杂交通环境中的纪录, 展示了我国无人驾驶技术向世界先进水平的迈进。同年, 军事交通学院研发的无人车在京津高速完成了百余公里测试, 最高时速达到 105 公里。2014 年, 百度发布了无人驾驶汽车研发计划, 采用智能驾驶技术, 并保留转向盘和踏板, 以实现人机共驾模式^[13]。2015 年, 宇通大型无人驾驶客车首次在高速公路上完成了 32.6 公里的尝试, 这也是国内第一次进行大型客车的无人驾驶试验。2016 年, 长安无人驾驶汽车六天内跨越多个省市, 完成了超 2000 公里上路测试。

2019 年，华为积极布局无人驾驶领域，与多家车企合作，其推出极狐阿尔法 S 能够实现高精度的环境感知和智能决策，在城市道路和高速路况下都展现出良好的自动驾驶性能。2020 年，百度 Apollo 持续升级，用户可通过手机 APP 预约乘坐 Apollo 无人驾驶出租车。2021 年，小鹏汽车在其旗舰车型上搭载了先进的自动驾驶辅助系统，能够根据高精度地图和传感器数据，实现自动变道、超车等操作，在国内一线城市的实际道路测试中表现出色。2022 年，毫末智行发布了自动驾驶生成式大模型，助力其无人车产品在复杂场景下的决策能力大幅提升。2023 年，大疆车载推出的智能驾驶解决方案被多家车企采用，其基于视觉感知的自动驾驶系统在一些车型上实现了精准的目标识别和路径规划，展现了国内无人驾驶技术在视觉感知领域的优势和创新成果，为无人驾驶技术的多元化发展提供了新的方向和动力。

1.2.2 目标检测算法研究现状

在无人驾驶车辆的运行过程中，环境中的交通标志对车辆的安全行驶至关重要。目前，交通标志检测主要通过对运行视频中关键帧进行分割，再利用特征匹配实现标志识别。随着智能算法的不断革新，各类目标检测算法相继问世，并在智能安防、交通管控、物流监测及无人驾驶等领域得到了广泛应用。

（1）传统目标检测算法

传统目标检测算法主要借助目标物体的内在特性，像颜色、形状以及纹理等来实现识别功能^[14]。如图 1-1 所示，在这一过程中，图像预处理环节至关重要，涉及去噪、增强、色彩空间转换和颜色直方图统计等操作。候选框生成是通过在待检测图片上滑动固定大小的窗口，将窗口内的子区域作为候选区域，并在这些候选区域内进行特征提取。分类器则负责判断提取到的特征是背景还是目标对象，如有多种目标类型，还需进一步区分其类别。最终识别出包含目标对象的候选框，但这些候选框会有很多，容易重叠在一起，这时就需要通过非极大值抑制（NMS）去除重叠度较高的冗余框，得到最终的目标位置或类别。

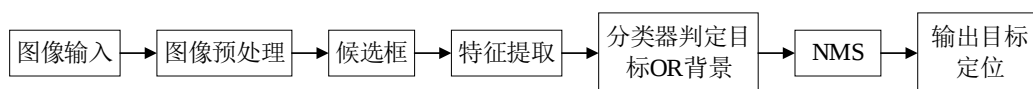


图 1-1 传统目标检测算法基本流程图

Li 等人^[15]提出了一种结合颜色不变量的图像分割和基于方向梯度金字塔直方图(Pyramid Histogram of Oriented Gradients, PHOG)特征的形状匹配检测技术。首先,通过提取图像的颜色恒常性特征,并利用这些特征进行聚类并且分割成不同的区域,有效地识别出潜在的交通标志区域,然后利用 PHOG 特征描述这些区域的形状特性,最后通过支持向量机进行分类,以确认是否为真正的交通标志。这种方法具有较高的召回率和良好的鲁棒性,但该算法处理时间长,存在一定的局限性。Xu 等人^[16]提出了一种结合自适应颜色阈值分割和形状对称假设检验的检测技术。首先利用直方图累积分布函数动态确定颜色阈值,并通过最大-最小归一化抑制高亮区域和背景干扰,其次将阈值图像的高光形状特征转化为连通域特征向量,采用统计假设检验算法,高效提取交通标志的感兴趣区域。该方法能准确识别交通标志,但实时性差。Hechri 等人^[17]通过模版匹配技术实现了对交通标志的识别。通过设置与交通标志大小一致的滑动窗口,排除非交通标志的干扰因素。但是一些交通标志的形状和大小存在差异,该方法鲁棒性表现欠佳。

(2) 单阶段目标检测算法

单阶段目标检测是目标检测算法中的重要类别,典型算法有 SSD^[18](Single Shot MultiBox Detector)和 YOLO^[19](You Only Look Once)等。其核心特点在于单次检测就能输出最终结果,检测过程中不存在候选区域这一概念。在执行检测任务时,对于输入的画面,网络会直接在画面中输出包含目标的检测框、目标种类以及目标的置信度。

与两阶段目标检测相比,单阶段目标检测在速度方面具有优势,能够快速处理图片并给出检测结果,特别适用于实时性要求较高的场景,如实时视频监控、无人驾驶中的实时路况感知等。然而,这种快速检测是以一定的精度损失为代价的,其精度相对两阶段目标检测算法略低。

YOLO 算法由 Redmon 等人^[20]首次提出,他将目标检测问题转化为回归问题处理,与传统的候选区域选择不同,YOLO 采用预设锚框的方式进行目标检测,这大大简化了计算过程并提高了检测速度。Chu 等人^[21]开发了 TRD-YOLO 模型,该模型结合了全局特征提取和多分支轻量化检测头,但对于特殊天气下的交通标志检测鲁棒性较差。Dewi 等人^[22]提出了一种基于空间金字塔池(Spatial Pyramid Pooling, SPP)的 YOLO 目标检测方法,由于 SPP 处理需要更多的时间,导致实时

性较差。郝等人^[23]提出了一种改进 YOLOv5 的手势识别算法,该算法通过融合生成对抗网络进行特征修复、Swin Transformer 增强全局感知能力,提高了手势检测的精度和鲁棒性,但检测速度有所下降。Chen 等人^[24]针对复杂场景下的船舶检测,提出了一种基于 YOLOv7 的 CSD-YOLO 检测模型。这一模型在复杂场景中展现出了卓越的检测性能,但其较高的模型复杂度也是不可忽视的一点。

(3) 两阶段目标检测算法

两阶段目标检测算法是一种精度较高但较为复杂的方法。它通过卷积神经网络提取特征,构建了包含候选区域网络和目标区域检测网络的两阶段架构来实现目标检测任务。

在第一阶段,即候选区域网络阶段,其核心操作是在输入的待检测图片上生成大量的候选区域。这一过程通过特定的算法和网络结构,依据图片中的像素信息、颜色分布、纹理特征等多方面因素,初步筛选出可能包含目标的区域。随后,对这些初步确定的候选框进行修正,通过调整候选框的位置、大小等参数,使得框内所包含的特征更具代表性,从而提高后续分析所使用特征的质量。

进入第二阶段,也就是目标区域检测网络阶段,主要任务是对第一阶段得到的候选区域进行深入处理。先对候选区域进行分类操作,利用训练好的分类模型判断每个候选区域中目标所属的类别。在此过程中,会进一步根据分类结果和目标的特征分布,再次精确地修正候选框,使最终输出的候选框能够更加精准地框定目标物体。最后,输出包含目标类别信息以及目标在图片中精确位置(以矩形框坐标形式表示)的结果。

Girshick 等人^[25]在研究中提出了 R-CNN 算法,被认为是两阶段检测算法的先驱。基于此,研究者们持续优化并推出了更高效的算法,如 Fast R-CNN^[26]、Faster R-CNN^[27]和 Mask R-CNN^[28]等。Cao 等人^[29]提出了一种改进的稀疏 R-CNN 算法,该算法通过优化主干网络和引进 GAP(Global Average Pooling)操作以及全连接层,保持较低的计算成本,实现了更出色的性能。Zhang 等人^[30]提出了一种多尺度级联 R-CNN 算法,使用多尺度关注技术,有效应对了现实交通场景中相似交通标志易误检的挑战。Qiao 等人^[31]提出了一种解耦更快的 R-CNN 网络结构,引入用于多级解耦的梯度解耦层和用于多任务解耦的原型校准块来扩展更快的 R-CNN。马等人^[32]提出了一种改进的 Mask R-CNN 舰船目标检测方法,该算

法融合注意力机制、特征金字塔网络及优化的特征提取策略，在舰船目标检测任务中实现了较高的检测精度和良好的泛化能力。然而，该方法计算成本较高，推理速度较慢，并且在密集目标场景下仍存在优化空间。

1.2.3 路径规划算法研究现状

近年来，随着科技日益进步，移动机器人在日常生活中的应用变得越来越广泛。这些机器人不仅能够胜任繁重且重复的工作，还能在紧急情况下代替人类完成一些危险任务。然而，为了实现任务目标，机器人必须准确找到通往目的地的路线，这就涉及到路径规划。移动机器人的路径规划就是在运行过程中对环境进行建图，将环境信息转化为数字数据，并利用路径规划算法设计出一条既最短又能避开障碍物的安全路径，从而确保机器人能够顺利到达目标点。

传统路径规划算法包含模拟退火算法^[33]、人工势场法^[34]、模糊逻辑算法^[35]、禁忌搜索算法^[36]等类别。模拟退火算法能够在一定程度上避免陷入局部最优解，通过随机接受劣解的机制实现全局搜索，适用于复杂的优化问题，但存在计算量较大、收敛速度慢等问题。Shi 等人^[37]提出一种改进的模拟退火算法。通过引入初始路径选择方法（根据特定规则和概率选择路径点，概率与地图大小相关）和删除操作（删除路径中不相邻但距离较近的点），使算法在动态环境下计算时间大幅缩短，路径规划性能显著提升，在静态和动态环境中均能找到接近最优解的路径，且路径长度相比其他算法更短；Pandey 等人^[38]针对移动机器人在存在不同形状静态障碍物环境中的路径规划与避障问题，采用模糊逻辑控制方法进行改进。通过建立 TS(Takagi-Sugeno)型模糊推理系统，定义输入变量（前、左、右障碍物距离）、输出变量（左右电机速度）及各自的三角模糊隶属函数（输入变量为“近”“中”“远”，输出变量为“快”“中”“慢”）来控制机器人运动，实现了机器人在未知环境中安全避障并成功到达目的地，且模糊推理系统能有效控制机器人导航行为。

基于自然界启示或生物进化形成的智能优化算法包含蚁群算法^[39]、神经网络算法^[40]、粒子群算法^[41]、遗传算法^[42]等类型。神经网络算法具有强大的学习能力，能够适应复杂多变的环境，通过对大量数据的学习可以处理各种非线性关系，从而实现高效的路径规划。Liu 等人^[43]提出一种基于粒子群优化训练模糊神经网络

络的算法。通过设计新的惯性权重和学习因子更新规则,改进粒子群优化算法,克服其易陷入局部极值和效率低的问题,利用粒子群训练模糊神经网络,调整其参数,设计简化的网络结构,以解决模糊神经网络收敛慢和计算量大的问题。实验表明,粒子群优化训练的模糊神经网络算法在路径规划上更优,路径更平滑、更短,收敛速度更快且稳定性强,在不同障碍物比例和复杂网络环境下适应性更好;Tuncer 等人^[44]对遗传算法中的变异算子进行了优化,在变异过程中不仅考虑变异节点,还全面检查该节点附近的所有可行节点。在此基础上,依据总路径的适应度值来精心选择节点,从而有效避免了传统随机逐个选择可能带来的非最优解接受问题。这种方法不再单纯依赖于变异节点的运动方向进行判断,而是采用了更为全面和精准的评估策略。实验结果表明,经过这种改进后的方法,在找到最优路径的次数上远超其他方法,同时在平均适应度值、平均迭代次数以及收敛速度方面也均展现出显著优势。

基于随机搜索原理的路径规划算法有迪杰斯特拉算法^[45]、A*算法^[46]、快速扩展随机树(Rapidly-Exploring Random Tree, RRT)算法^[47]等。A*算法能够在复杂环境中高效地寻找最优路径,原理简单易实现,但在障碍物多或地图大时,计算复杂度增加,路径不够平滑,不利于运动控制。针对 A*算法所存在的问题,Zhang 等人^[48]提出基于自适应邻域搜索和转向成本的改进算法。根据周围障碍物信息自适应选择搜索邻域,突破 8 邻域搜索节点限制,并且在评估函数中引入转向成本,使算法选择最优子节点时兼顾直线成本和转向成本。Li 等人^[49]也对 A*算法进行了改进,通过引入双向交替搜索(Bidirectional Alternating Search, BAS)策略,提升搜索效率。采用指数衰减加权启发式函数,克服了 BAS 算法在遇到垂直于路径的障碍物时计算时间增加的缺点。此外,还设计了一个路径节点过滤函数,该函数能够有效地剔除路径中的冗余节点,并通过贝塞尔曲线拟合使路径平滑,这些改进措施共同作用下,不仅大幅提升了路径规划的速度,还缩短了路径长度,减少了转弯角度。

1.3 论文主要研究内容与章节安排

本研究着重讨论了基于交通标志检测的无人车路径规划方法,通过改进目标

检测和路径规划算法，实现无人车在遵守交通规则的前提下，规划出一条长度最短且能规避障碍的平滑行驶路线，本文一共有六章，具体研究内容如下：

第一章：主要阐述了研究课题的背景与意义，系统梳理了相关技术在国内外的研究现状，并详细介绍了论文的整体框架。

第二章：详细阐述了本研究涉及的理论基础。首先介绍了卷积神经网络，并深入解析了其各个层的工作原理。然后通过对智能小车运动学模型的分析，为路径规划提供了运动约束依据。最后探讨了栅格地图、拓扑地图等常见环境建模方法。这些理论内容为后续目标检测算法的改进和路径规划算法的设计奠定了基础。

第三章：首先详细介绍了 YOLOv8s 目标检测算法，分析其基本原理和网络结构，并针对现有交通标志检测技术在小目标检测、复杂背景适应性方面的不足，提出了一种改进模型。该模型通过引入轻量级 RepGhost 模块降低计算复杂度，新增小目标检测层提升检测能力，融入 CBAM 注意力机制增强目标聚焦能力，并采用 MPDIoU 损失函数优化模型收敛性能。最后，在 TT100K 数据集上的实验结果表明，改进后的 YOLOv8s 算法在交通标志检测任务中进一步提升了检测准确性。

第四章：本章介绍了蚁群算法与人工势场法的基本概念，并解析了各参数的作用。针对传统路径规划算法在局部最优、避障能力及交通规则融合方面的不足，提出了一种改进蚁群算法与人工势场混合的路径规划方法。改进措施包括非均匀初始信息素分布、自适应信息素更新机制、惩罚系数调整及人工势场避障策略，同时采用三次 B 样条曲线优化路径平滑性，并将交通标志约束融入路径规划过程。仿真实验结果表明，该改进算法在路径长度、运行时间及复杂环境适应性方面均具有显著优势。

第五章：首先阐述了无人车的硬件和软件构成及其通信方式，随后介绍了实验场地的设置情况与实验的具体流程。为验证所提方法的有效性与可信度，设计了左转和右转交通标志的测试场景，将改进后的 YOLOv8s 交通标志检测算法和蚁群混合人工势场路径规划算法部署到 HW-Autocar 无人车平台进行测试。

第六章：对全文的研究工作进行了总结，分析了研究中的局限性，并提出了未来的研究方向。

第 2 章 相关背景知识介绍

本章旨在为无人车路径规划和交通标志检测研究奠定理论基础。主要介绍了卷积神经网络的结构与作用、智能小车的运动学模型以及环境建模方法，为后续算法设计与实验验证提供必要的理论技术支撑。

2.1 卷积神经网络

卷积神经网络^[50](Convolutional Neural Networks, CNN)是一种前馈神经网络，具有深度结构并包含卷积计算，是深度学习领域的典型算法之一。

在 20 世纪 60 年代，Hubel 和 Wiesel 通过实验发现，大脑中的部分神经元对特定方向的边缘敏感，并且这些神经元以柱状结构共同协作以产生视觉感知，这为 CNN 的发展奠定了生物学基础。1998 年，LeCun 提出了 LeNet^[51]模型如图 2-1 所示，该模型通过梯度反向传播算法进行监督训练，它将原始图像经过卷积层和下采样层的交替处理，转换为特征图，最后由全连接神经网络进行分类，在手写体数字分类任务上取得了显著效果。

2012 年，AlexNet 在 ILSVRC 竞赛中获得第一名，其优势显著，使得卷积神经网络受到了广泛关注。此后，众多新模型如 VGG^[52]、GoogLeNet^[53]、ResNet^[54]等不断涌现，同时 CNN 与传统算法相融合，迁移学习方法也被引入，这极大地拓展了 CNN 的应用领域，如图像识别、视频行为识别、自然语言处理、语音识别等。

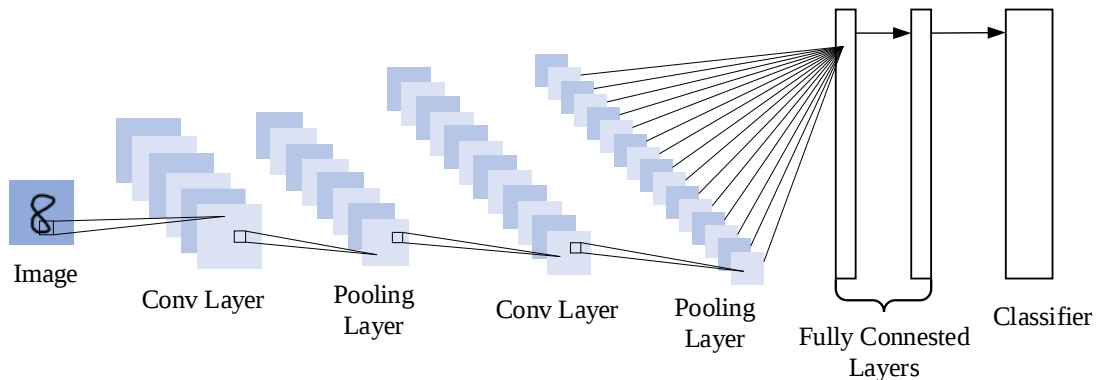


图 2-1 LeNet 模型结构图

2.1.1 输入层

输入层是卷积神经网络的基本组成部分，主要用于接收输入数据，通常不进行任何计算和特征提取。输入层的设计会直接影响网络的性能和精度，特别是在处理图像数据时，输入数据的预处理非常关键。输入层通常由多个节点组成，节点数与输入数据的维度相同，这些节点代表输入数据的特征或属性。常见的输入层设计包括直接使用原始像素数据、采用预训练模型提取的特征输入，以及通过数据增强技术扩展输入数据的多样性。一个合理的输入层设计可以显著提高网络的泛化能力和鲁棒性。

2.1.2 卷积层

卷积层可以通过输入的数据获取其局部特征。通过卷积操作，卷积层能够有效识别图像中的基本特征，如边缘、角点和纹理等。这使得卷积神经网络在处理图像、语音和视频等数据时，能够高效地学习数据的空间结构。卷积层的核心操作是卷积，卷积是通过一个小矩阵（称为卷积核或滤波器）与输入数据进行逐元素相乘并求和的运算。卷积核在输入数据上滑动，从而提取局部区域的特征。

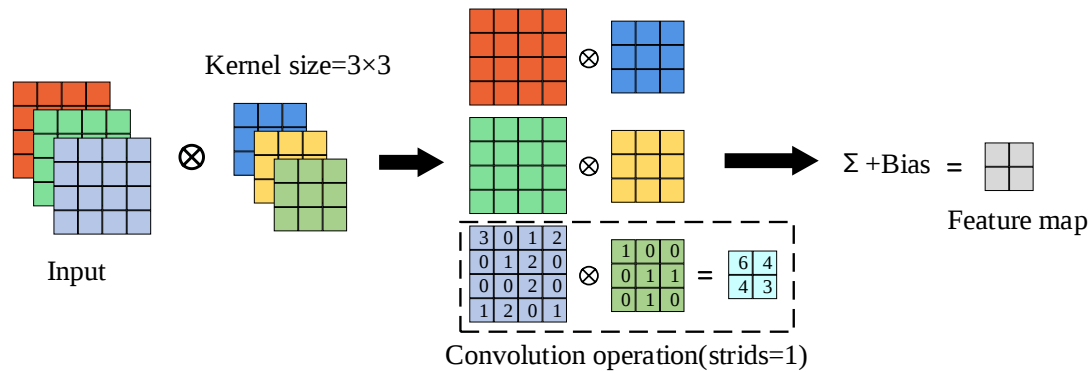


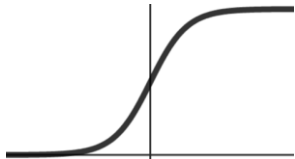
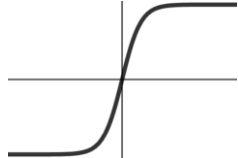
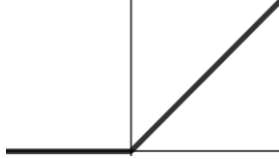
图 2-2 卷积计算过程示意图

如图 2-2 所示，输入特征图的通道数与卷积核的通道数保持一致，卷积核通过滑动方式依次与输入特征图的对应通道局部区域进行卷积运算得到该区域的特征值。根据设定的步长，卷积核在特征图上从左到右、从上到下滑动，覆盖所有可能的位置，生成一个单通道的结果矩阵。最终，将所有通道的卷积结果相加，再加上偏置，得到输出的单通道特征图。

2.1.3 激活函数

激活函数^[55]的作用是对卷积操作产生的输出进行非线性变换，从而引入非线性特性，以增强模型表达能力，控制梯度值以避免梯度爆炸或消失问题，以及稳定训练过程并提高网络的拟合能力，使网络能够学习和表示复杂的非线性关系。常见的激活函数如表 2-1 所示。

表 2-1 激活函数

激活函数	表达式	激活函数图像
Sigmoid	$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}$	
Tanh	$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	
ReLU	$\text{ReLU}(x) = \max(0, x)$	

Sigmoid 函数^[56]是一种常见的激活函数，主要作用是将输入值映射到(0,1)的范围内，从而将任意实数转化为概率形式。Sigmoid 函数的形状是一个平滑的 S 型曲线，这使得它在处理二分类问题中非常适用。它具有良好的可微性，适合反向传播计算梯度，但也存在梯度消失问题，尤其是在深层神经网络中，可能导致模型训练缓慢或停滞。

Tanh 函数^[57]又称为双曲正切函数，形状也是一种平滑的 S 型曲线，属于 Sigmoid 函数的变体。当输入为 0 时，输出也为 0，输入较大时，输出趋近于 1，输入较小时，输出接近于-1。与 Sigmoid 函数相比，Tanh 函数的输出具有零均值特性，有助于加快神经网络的收敛。

ReLU 函数^[58]是一种常用激活函数，其特点是输入大于 0 时输出为输入值，输入小于等于 0 时输出为 0。ReLU 函数简单高效，具有非线性特性，同时能够缓解梯度消失问题，使深层神经网络更容易训练。相比于 Sigmoid 和 Tanh 函数，ReLU 的计算开销更低，并且能在正区间内保持梯度稳定。

2.1.4 池化层

池化层是卷积神经网络中用于对特征图进行降维的关键部分，其作用是减少计算量和参数规模，同时保留关键信息，增强模型对位移、缩放和形变的鲁棒性。池化层通过滑动固定大小的窗口对局部区域进行操作，生成更小的特征图，从而压缩信息。池化层还能够有效防止过拟合，通过提取局部区域的统计特征，保留重要特征的同时丢弃冗余信息。

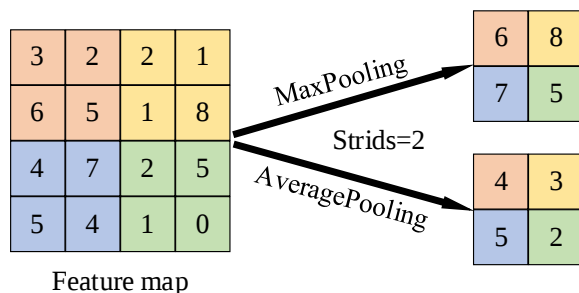


图 2-3 池化示意图

池化操作与卷积过程类似，通过滑动固定大小的窗口在输入特征图上逐区域进行扫描，对每个窗口内的像素值进行降维处理。最大池化在窗口区域内选取最大值作为输出，能够提取局部区域的显著特征；而平均池化则计算窗口内像素值的平均值，保留区域的整体信息。图 2-3 中采用步长为 2 的池化操作，对输入 4×4 的特征图进行降维处理。最大池化在每个 2×2 的窗口中选取最大值，结果为 2×2 的输出特征图[6,8,7,5]；平均池化在每个窗口中计算平均值，结果为 [4,3,5,2]。池化操作的核心作用是通过降维保留关键信息，减少计算量和参数规模，同时提高模型的鲁棒性。

2.1.5 全连接层

全连接层是网络的最后几层，其作用是将前面提取到的特征整合起来，用于实现特定的任务。全连接层通过与每一个输入特征点建立权重连接，并结合激活函数，对特征进行加权求和，从而生成高层次的特征表示。全连接层区别于卷积层中的局部连接，每个神经元与前一层的所有神经元都完全连接，因此能够综合前面所有提取到的特征，为最终输出提供全局信息。它能够将卷积层和池化层提取的局部特征融合成全局特征，减少特征位置对任务结果的影响，同时将特征向

量映射到输出层的神经元，完成分类或回归任务。然而，由于全连接层参数量较大，训练时计算资源需求较高，实际应用中往往需要优化和减少参数，以提高效率并降低过拟合风险。

2.2 智能小车运动学模型

本实验研究平台所用的智能小车为 HW-Autocar 移动机器人，如图 2-4 所示。



图 2-4 HW-Autocar 智能车

这款智能车采用了华为昇腾 310AI 处理器，并集成了强大的硬件和软件平台，适合用于自动驾驶和机器人技术的研究与开发。车辆支持 ROS^[59]开发平台，可以实现快速的 4m/s 自动驾驶，具有激光雷达地图构建、全局和局部路径规划等功能。此外，车载系统是开放源代码，支持二次开发，并且能够用于无人驾驶算法的验证。该车产品参数如表 2-2 所示。

表 2-2 HW-Autocar 智能车参数

项目	参数
车身尺寸	56×35×23cm
主控制器	昇腾 310AI 处理器
底盘	阿克曼底盘
IMU	姿态角动态精度 0.5 度，航向角动态精度 2 度
激光雷达	360 度全方位扫描，10 赫兹自适应扫描频率，测量距离 25 米
里程计记录器	臂长 15cm，同步轮直径 68mm
摄像头	720P，60 帧/秒，工业级 100 万像素
软件系统	Ubuntu 22.04，ROS2_humble

为了制定既符合车辆物理特性又满足安全要求的控制策略，必须构建无人车

的运动学模型，而该模型需满足以下条件：

(1) 小车的车轮是理想的刚性车轮，且与地面之间的摩擦力是恒定的，车轮转动时没有打滑现象。

(2) 小车的车体是刚性物体，车辆各部分之间没有形变或弹性变化。

(3) 小车的运动只受车轮转动的控制，不考虑动力学因素，如车速变化带来的惯性效应。

(4) 车辆在运动过程中保持稳定的行驶状态，即车轮的转速与车辆的实际速度相匹配。

本实验所用的 HW-Autocar 移动机器人阿克曼底盘，阿克曼底盘 (Ackermann Steering) 是一种广泛应用于车载机器人和自动驾驶系统的底盘设计。它的结构与传统的汽车类似，采用前轮转向，后轮驱动。阿克曼底盘的转向几何设计确保了车轮在转弯时都能按照理想轨迹运动。内侧车轮转向角较大，外侧车轮转向角较小。这样可以使得四个车轮始终指向同一个转弯中心点，从而减少轮胎的磨损，避免打滑，并提高行驶的稳定性和。如图 2-5 所示为阿克曼底盘转向运动学模型。

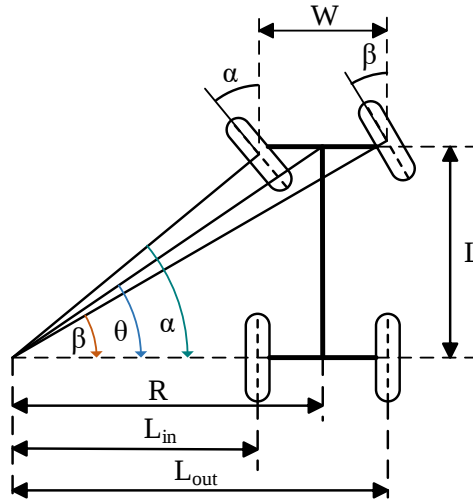


图 2-5 阿克曼底盘转向运动学模型

图中 α 、 β 分别为左、右前轮的转向角， L 为车体前轴与后轴之间的距离， W 为车体左前轮与右前轮之间的距离， R 为车辆的转弯半径， θ 为车辆的航向角，其计算公式如式(2-1)、式(2-2)所示， α 与 β 的关系满足式(2-3)。

$$R = \frac{W}{2} + \frac{L}{\tan \alpha} \quad (2-1)$$

$$\theta = \arctan\left(\frac{L}{R}\right) \quad (2-2)$$

$$\frac{\tan \alpha}{\tan \beta} = \frac{R + \frac{W}{2}}{R - \frac{W}{2}} \quad (2-3)$$

假设车辆的行驶速度为 v ，此时车辆的当前位置可以通过其线速度和航向角的变化进行更新，计算公式如式(2-4)、式(2-5)所示。

$$\frac{dx}{dt} = v \cdot \sin \theta \quad (2-4)$$

$$\frac{dy}{dt} = v \cdot \cos \theta \quad (2-5)$$

其中， x 和 y 分别表示车辆在世界坐标系中的位置。

2.3 环境建模

在路径规划中，环境地图至关重要。无人车在规划路径前，必须采集周围环境数据，这些数据可以通过激光雷达、相机等多种传感器获取，并经过处理后转化为适合路径规划的环境地图。

为了更全面地构建环境地图，智能车还需要考虑动态障碍物的检测与跟踪，以及环境中的交通标志、信号灯等元素的识别。这些信息不仅有助于提高路径规划的准确性，还能增强智能车在复杂环境中的适应能力。通过融合多源传感器数据，环境地图可以更精确地反映现实世界的复杂情况，从而为智能车提供更可靠的导航支持。

2.3.1 环境地图介绍

在路径规划中，环境地图的精确度至关重要，因此应根据不同环境特点采用相应的建图方法。在智能车路径规划中，常见的地图类型有三种^[60]：

(1) 拓扑地图(Topological Map)：通过将地图中的重要节点（例如交叉点、路口等）连接起来，表示不同节点之间的关系或路径。这些节点之间的连通性通过图的形式表示，适用于描述复杂网络或路径的结构。

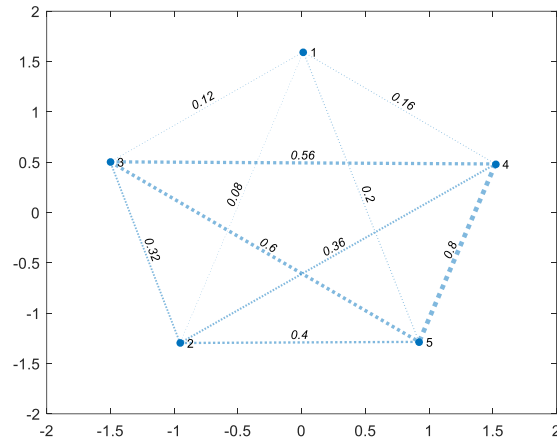


图 2-6 拓扑地图

拓扑地图通过节点和连接它们的边来表示空间的结构，描述环境中各个重要地点之间的关系。图 2-6 中的每个蓝色圆点代表一个节点，这些节点通常代表环境中的关键位置（如交叉口、站点、目的地等）。节点之间的虚线表示两两之间的连接关系，数字则代表连接的权重（例如距离、时间或成本等）。

（2）尺度地图(Metric Map): 是将真实世界按比例缩小，每个地图点对应于现实世界的一个特定位置，如经纬度坐标、局部坐标系中的位置等。其能精确地表示环境的几何形状，保留了空间的真实比例和位置，但对于大范围的环境，其地图数据也较为庞大，可能会带来较大的计算负担。

（3）语义地图(Semantic Map): 在尺度地图的基础上，添加了环境中的语义信息（如路口、交通标识、障碍物等），使得地图不仅仅是空间位置的表示，还包含了对空间元素的理解和描述，但构建和维护语义地图的难度较大，需要大量的人工标注和数据处理。

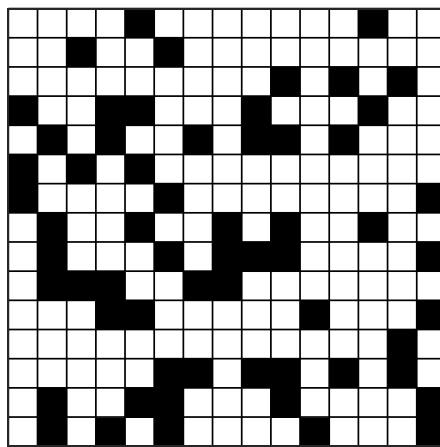


图 2-7 栅格地图

本文采用最常见的栅格地图进行路径规划研究。如图 2-7 所示，栅格地图将环境划分为若干个均匀的网格单元，每个网格表示现实空间中的一个区域，每个网格中可以包含环境信息（如是否有障碍物、自由空间或未知区域等）。此种地图结构简洁、易于计算，适合用于动态路径规划和实时导航。

栅格地图最早由 Elfes 和 Moravec 提出^[61]，其核心思想是将真实环境按照一定比例离散化为多个栅格，从而表达地图信息。无人车通过相机采集图像与激光雷达扫描，获取周围环境信息，并利用数据融合技术将这些信息构建为栅格地图。栅格地图的优势在于能够较为完整地保留环境信息，便于后续地图重建与维护。然而，其缺点在于，当环境规模较大时，栅格地图的构建可能面临效率低、成本高的问题。

2.3.2 地图栅格化

地图栅格化是一种将连续的真实环境离散化为规则网格单元（栅格）的建模方法。每个栅格代表环境中的一个局部区域，通过数值或概率表示该区域的属性（如障碍物存在性、空闲状态等）。其核心思想是通过空间离散化和概率量化，将复杂环境简化为计算机可处理的数据结构。如图 2-8 所示为地图栅格化的流程图。

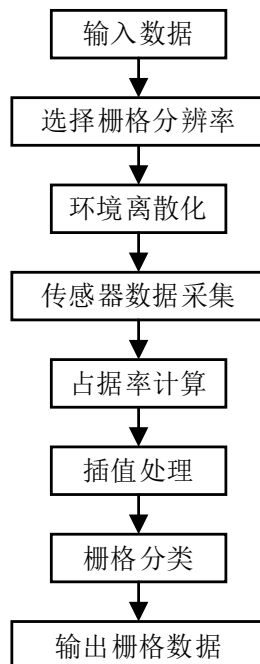


图 2-8 地图栅格化流程图

地图栅格化的过程是将连续的空间数据（如矢量数据或影像数据）转换为规则栅格的过程。首先，基于预设分辨率将环境均匀划分为二维栅格矩阵，并建立全局坐标系实现空间对齐。接着，通过激光雷达、深度相机等传感器实时采集点云或深度数据，经坐标系转换、噪声滤波和运动畸变校正后，利用贝叶斯滤波对多帧数据进行融合，动态计算每个栅格的占据概率，如果栅格化过程中存在数据空缺，可能会使用插值方法（如最近邻插值、双线性插值等）填补缺失的栅格值，以确保数据的连续性和空间精度。随后区分自由区域、障碍区域及未知区域，进一步结合语义信息，将栅格细化为六类：自由栅格、已知静态障碍、未知静态障碍、未知动态障碍、起始点及目标点，并构建八叉树结构优化存储效率。最后，生成的栅格数据通常以 GeoTIFF 或 ASCII Grid 等栅格数据格式保存。

在智能车路径规划中，栅格大小会影响路径规划精度，同时也受智能车尺寸制约。栅格分辨率越高，地图对环境的描述越精细，路径规划结果也会更加准确。但较高的分辨率意味着更多的栅格单元，从而增加了计算量，可能导致路径规划的实时性降低。因此，在栅格地图的精度和计算效率之间需要做出平衡，既要保证路径规划的精度，又要确保系统能够实时响应。选取合适的栅格大小，是路径规划中优化计算资源和提升路径精度的关键因素。

2.4 本章小结

本章介绍了无人车路径规划和交通标志检测的相关理论基础。首先，介绍了卷积神经网络的基本原理及其在目标检测中的应用，重点讲解了 CNN 的各个层次结构以及其在处理图像数据中的作用。接着分析了智能小车的运动学模型，为路径规划提供了运动约束依据。最后，介绍了环境建模的基本方法，特别是栅格地图的构建和栅格化过程，这些为路径规划提供了基础支持。

第 3 章 基于改进 YOLOv8s 的交通标志检测

交通标志检测是无人车安全导航的关键，第二章介绍了 CNN 等相关理论支持。本章针对 YOLOv8s 模型，提出了一种改进方案以提升检测性能：引入轻量级 RepGhost 模块以降低计算复杂度，新增小目标检测层以增强小目标识别能力，融入 CBAM 注意力机制以聚焦复杂背景中的目标，并采用 MPDIoU 损失函数优化收敛速度。最后，在 TT100K 数据集上的实验结果验证了改进模型的有效性。

3.1 YOLOv8s 模型

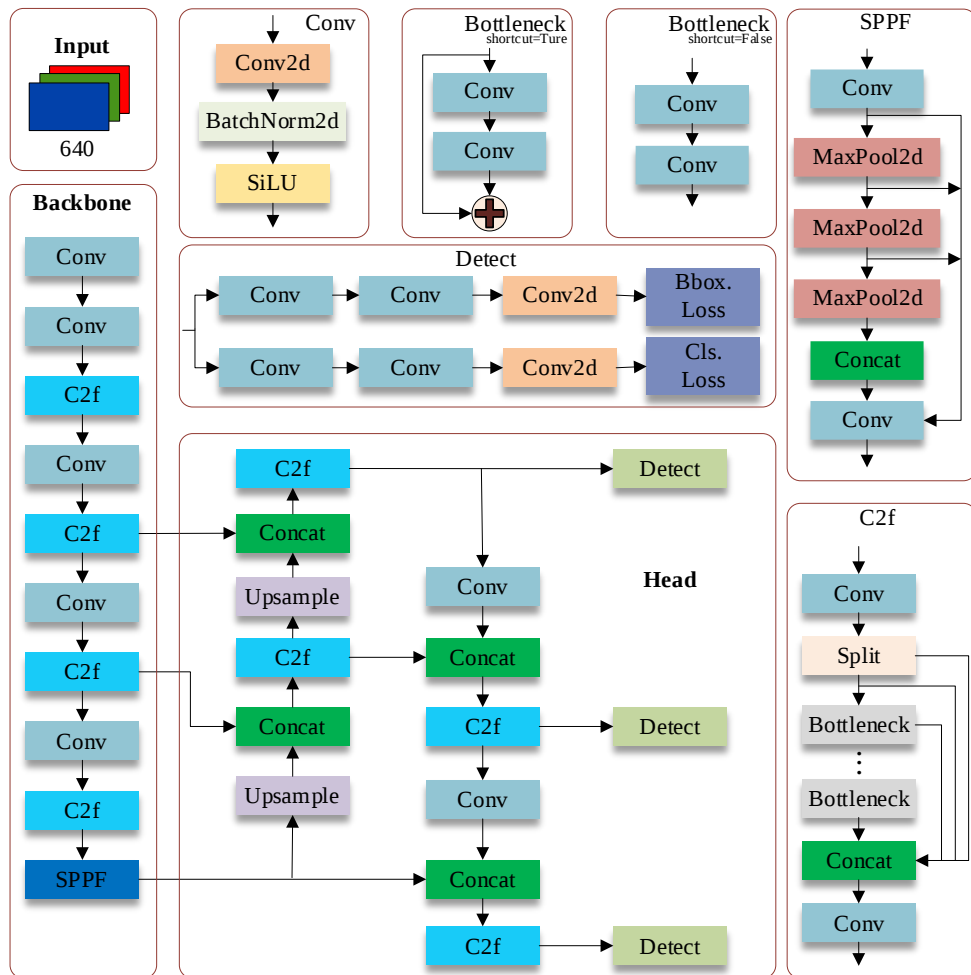


图 3-1 YOLOv8s 网络结构图

YOLOv8 是 Ultralytics 推出的 YOLO 系列目标检测算法，相较于其他的 YOLO 版本，YOLOv8 在网络设计上引入了更加高效的特征提取模块，并采用了

自适应锚框机制与动态标签分配策略,对于复杂交通环境下的标志识别效果较好。YOLOv8 的性能因网络复杂程度的不同而有所差异,不同模型展现出不同的性能特点,综合考虑精度和速度的需求,本文选择了 YOLOv8s 网络模型。该模型由四大部分构成:输入端负责接收图像数据,主干网络负责特征提取,颈部网络进行特征融合与增强,而输出端则负责预测并生成最终的目标检测结果。图 3-1 展示了 YOLOv8s 网络模型的结构示意图。

输入端主要有 Mosaic 数据增强、自适应锚框计算和自适应灰度填充。其中 Mosaic 数据增强功能在训练的最后 10 个 epoch 将会关闭,可以有效地提升精度。YOLOv8s 采用无锚框(anchor-free)的目标检测策略,无需设定预定义锚框,仅通过目标的中心点以及宽高在不同尺度特征图上进行回归,降低时间和计算资源消耗。

主干网络有 Conv、C2f、SPPF 等模块构成,主要用于对目标的特征提取。其中 C2f 模块是 YOLOv8s 网络的创新,主要用于残差特征的学习,该模块参阅了 YOLOv7 的 ELAN 规划思想,通过更多的分支跨层链接,使得网络实现了轻量化设计并丰富了梯度流信息。SPPF 模块采用空间金字塔池化技术,有效融合不同尺度的特征,将其整合到统一尺寸的特征向量中,以增强模型对多尺度目标的检测能力。

颈部网络由特征金字塔网络(Feature Pyramid Network, FPN)和路径聚合网络(Path Aggregation Network, PAN)构成,主要作用是整合高级和低级特征,形成一个特征金字塔。FPN 通过自顶向下的方法融合不同层次的图像特征,而 PAN 则从底部向上融合各层次的语义和图像特征。这一结合不仅融合了浅层的图形细节与深层的语义信息,而且为网络提供了更全面的特征信息,从而增强了检测效能。

输出端对分类与检测进行解耦,主要涉及损失计算与目标检测框筛选。损失计算包含分类与回归两个分支,无 Objectness 分支,其中分类采用 BCE Loss,回归使用 Distribution Focal Loss 和 CIOU 损失函数。

3.2 改进的 YOLOv8s 模型

针对当前交通标志识别面临的挑战,本文提出了一种改良的 YOLOv8s 模型

用于交通标志的检测。如图 3-2 所示，该模型的改进主要集中在以下四个方面：为了减少模型计算量，将轻量级的 RepGhost 模块和 Bottleneck 模块进行融合；为了更精准地检测道路交通标志中的小目标特征，增加了更小尺寸的检测层；同时，考虑到交通标志可能出现在复杂的环境背景中，在主干网络中融入了 CBAM 注意力机制，更好地聚焦于目标；为了加快网络模型的收敛速度，将原网络的损失函数替换成了 MPDIoU。

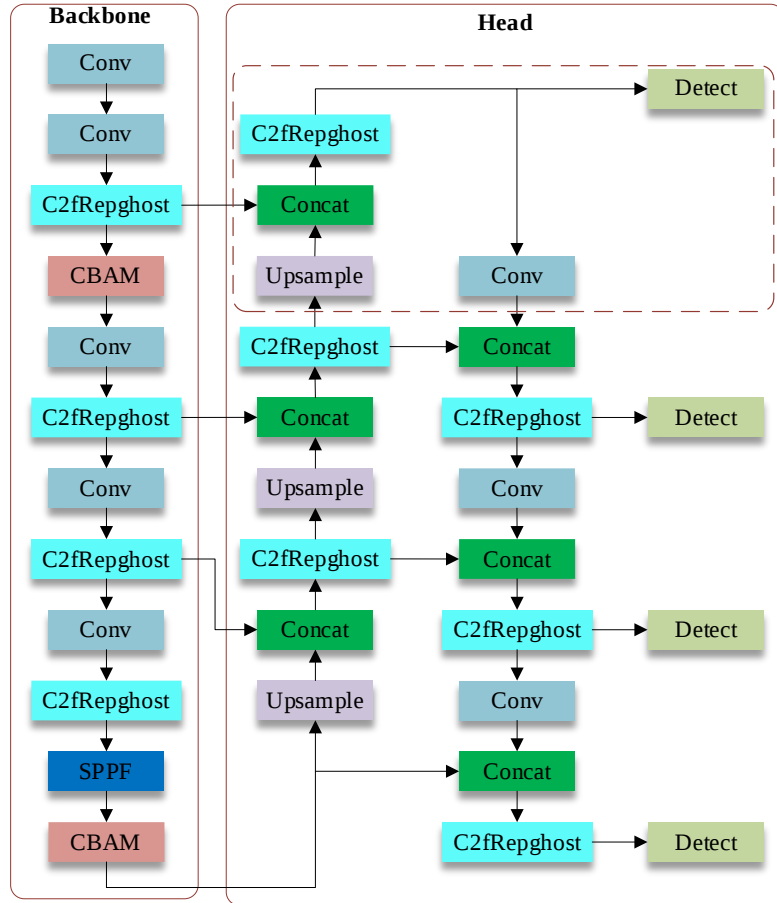


图 3-2 改进 YOLOv8s 网络结构图

3.2.1 网络结构轻量化设计

由于原 YOLOv8s 模型参数较大，计算复杂，难以在边缘设备上广泛部署，实现实时检测，因此，本文引入了 GhostNet 提出的 Ghost 模块来满足轻量级模型的需求。Ghost 模块首先利用少量的常规卷积产生部分特征图，随后在这些特征图上应用线性卷积以产生新的特征图，并通过 Concat 操作合成最终的特征图。Ghost 模块如图 3-3 所示，其中 ϕ 表示线性变换，是一种不含批归一化和非线性

激活的 3×3 卷积。

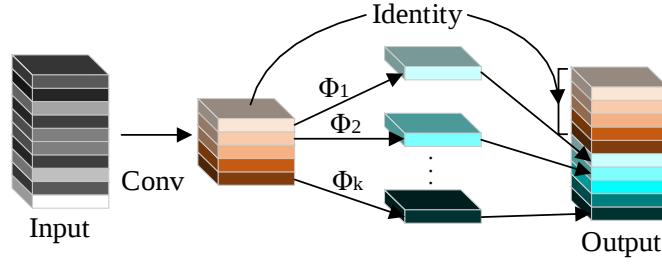


图 3-3 Ghost Module

如图 3-4 所示，RepGhostConv 模块采用两步特征提取策略：首先通过部分普通卷积核生成一部分特征图，随后在已有特征图基础上，利用 3×3 卷积核和步长为 1 的线性操作(cheap operation)生成剩余特征图，最后将两次得到的特征通过加法操作进行融合。RepGhostConv 模块删除了效率低下的连接符操作由加法运算符替代执行，提高了推理速度。

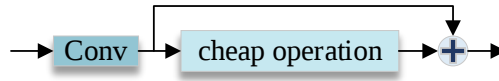


图 3-4 RepGhostConv 结构

RepGhostBottleneck 模块工作流程如图 3-5 所示，首先利用第一个 RepGhostConv 模块扩展通道数，实现特征的初步变换。随后，通过批量归一化和 SiLU 激活函数稳定训练过程并引入非线性，增强网络表达能力。接着第二个 RepGhostConv 模块减少通道数，使其与原始输入相匹配，便于后续融合。再次进行批量归一化，促进特征融合与网络收敛。最后，将处理后的特征图与原始输入特征通过残差连接进行相加，实现特征的有效整合。

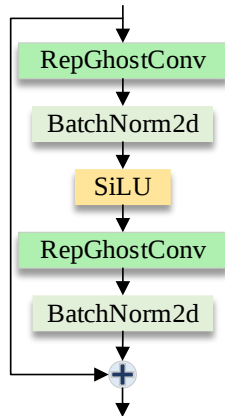


图 3-5 RepGhostBottleneck 结构

利用 Ghost 模块和 RepGhostBottleneck 的优势,设计了一个名为 C2fRepGhost 的轻量级特征提取结构。如图 3-6 所示,该结构首先通过一个卷积层,处理输入生成特征图,接着这些特征图被分割(Split),降低模型的计算成本,提高处理速度。分割之后的特征图进入两个连续的 RepGhostBottleneck 模块,该模块通过跨层特征融合和多分支梯度流方法对特征进行加工,增强特征的表达能力。然后将这些特征再次合并,整合不同层级特征的信息,增加特征的丰富性。最后一个卷积对合并后的特征图进行处理,输出最终的特征表示。

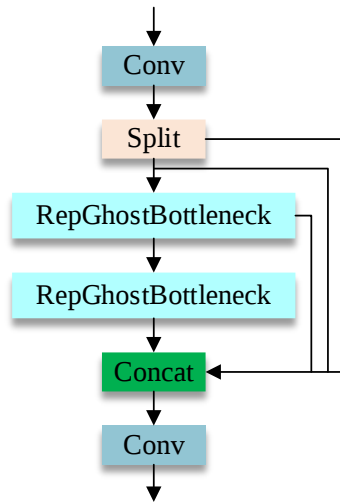


图 3-6 C2fRepGhost 结构

3.2.2 引入 CBAM 注意力机制

针对交通标志数据集中存在的交通标志偏小、位置边缘化以及多个标志紧密排列导致检测漏检等问题,本文将 CBAM 注意力模块添加到主干特征提取网络中。其作用是融合通道和空间两个维度的注意力机制,这种融合方式不仅强化了模型的特征提取能力,使得模型能够更精准地捕捉到关键信息,还进一步提升了结果的准确性。

CBAM 注意力机制巧妙地融合了通道注意力机制(channel attention)和空间注意力机制(spatial attention),协同学习交通标志图像中的关键局部细节信息。如图 3-7 所示,特征首先接受通道注意力机制处理,产生通道注意图 M_c ,其次将 M_c 和输入特征 F 做 element-wise 乘法操作,生成新特征 F' ,这个新特征 F' 随后被传递至空间注意力机制,生成空间注意图 M_s ,最后将 M_s 和 F' 做乘法,获得最终

生成的特征 F'' 。该过程可用式(3-1)表示：

$$\begin{aligned} F' &= Mc(F) \otimes F \\ F'' &= Ms(F') \otimes F' \end{aligned} \quad (3-1)$$

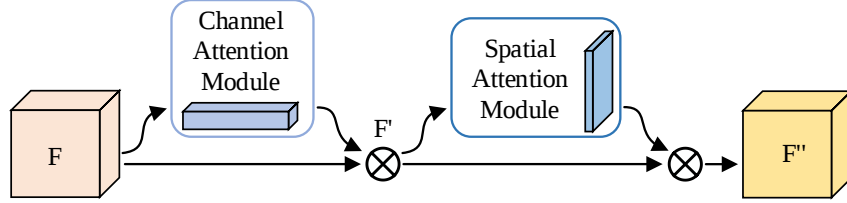


图 3-7 CBAM 注意力机制结构

通道注意力机制的结构如图 3-8 所示，输入层接收特征图 $F \in \mathbb{R}^{C \times H \times W}$ ，其中 H 和 W 分别表示输入特征图的高度与宽度， C 代表通道数。首先特征图经过池化操作以压缩空间信息，降低模型的复杂度，同时保留重要的特征信息。然后两层全连接层进一步整合并抽象提取的特征，MLP(Multi-Layer Perception)是一个多层感知机，它利用非线性激活函数将特征进行映射。再将生成的特征进行组合和 Sigmoid 函数归一化，最终得到融合特征各通道注意图 Mc 。该过程可以用式(3-2)表示：

$$\begin{aligned} Mc(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= \sigma(W_1(W_0(F_{avg}^c)) + W_1(W_0(F_{max}^c))) \end{aligned} \quad (3-2)$$

式中： $W_0 \in \mathbb{R}^{C/r \times C}$ 和 $W_1 \in \mathbb{R}^{C/r \times C}$ 是 MLP 两个隐藏层的权重， σ 表示 Sigmoid 激活函数。

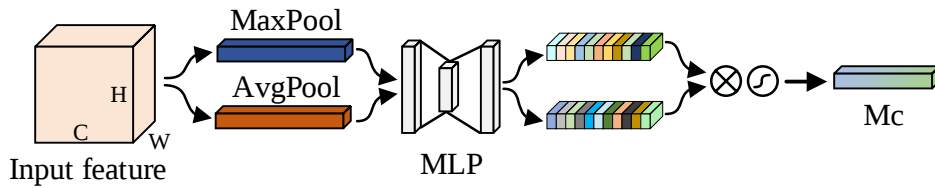


图 3-8 通道注意力模块

空间注意力机制结构如图 3-9 所示，输入层接收到的特征是由通道注意力模块融合的特征 F' 。首先特征经过两个池化层（最大池化层和平均池化层），这两个池化层对特征在通道维度上进行单通道池化操作，然后再进行通道级联融合特征，融合后的特征再经过一个 7×7 的卷积来过滤特征，最后得到特征空间注意力图 Ms 。该过程可用式(3-3)表示：

$$\begin{aligned}
 Ms(F) &= \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)])) \\
 &= \sigma(f^{7 \times 7}([F_{avg}^s; F_{max}^s]))
 \end{aligned}
 \tag{3-3}$$

式中： $f^{7 \times 7}$ 表示卷积核为 7×7 大小的卷积层。

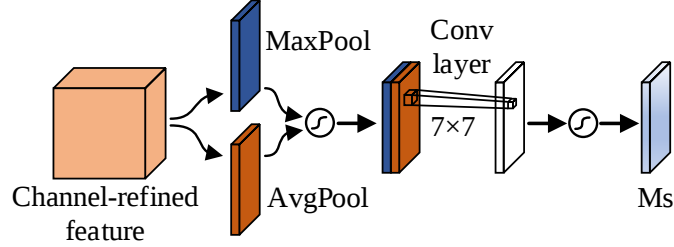


图 3-9 空间注意力模块

3.2.3 针对小目标增加预测头

原始的 YOLOv8s 算法中，输入图像经过不同比例的下采样处理，具体包括 8 倍、16 倍和 32 倍的下采样。这一操作旨在生成三种不同尺寸的特征图，即 80×80 、 40×40 和 20×20 ，这些特征图分别对应不同大小的目标检测任务。考虑到车辆需要在与交通标志有一定距离时就能进行识别，此时交通标志在图像中会显得较小，但是模型采用了较大的下采样倍数，经过多次下采样后，特征图中的目标信息变得非常稀少。这导致了原有的检测层无法有效地检测小目标，容易产生漏检、误检现象。

因此，本文在现有三层检测基础上增加了一个专门的小目标检测层，如图 3-2 中虚线框所示。该小目标检测层具有 160×160 的分辨率，以保留更多关于小目标的位置信息和细节特征信息，从而更有效地捕捉小目标的位置。该层主要由卷积操作、上采样步骤以及 C2fRepGhost 模块构成。首先，该检测层的输入由第六层特征提取网络提取的特征和第十五层特征提取网络经过上采样获得的特征相融合而成，这种融合操作的目的是提取出包含小目标深层语义信息的特征。融合后的特征图经过上采样过程，增加了特征图的尺寸，从而使其包含更多关于小目标的详细信息。随后，C2fRepGhost 模块处理上采样的特征图，进一步增强其语义表达，使特征图能够更精确地捕捉小目标的属性。经过上采样和 C2fRepGhost 处理，特征图融入了小目标的深层语义信息。随后，这一特征与网络中第三层的浅层位置特征相融合，生成了 160×160 尺度的融合特征层，它有效地结合了目标的语义特征和位置信息。这一结果通过两条路径输出：一路直接

作为检测头，专门用于识别尺寸较小的交通标志；另一路则经过卷积处理，再与未上采样的特征图结合，作为 Neck 的输入，进一步优化模型的检测能力。

3.2.4 MPDIoU 损失函数

在 YOLOv8s 中，损失函数由两部分组成：分类损失和回归损失。其中，分类损失采用了 BCE Loss，用于衡量锚框与对应标定分类的匹配程度；回归损失则采用了 DFL Loss+CIoU Loss，用于衡量预测框与真实框之间的差异。

CIoU 损失函数在描述边界框的重叠率时存在不确定性，同时忽视了不同样本的差异性。为了解决这些问题，本文引入了 MPDIoU 损失函数。MPDIoU 不仅考虑了边界框的交集区域、中心间距和宽高比例等因素，还额外权衡了目标的非交集部分。通过这种方式，MPDIoU 解决了 CIoU 损失函数在真实框与预测框没有交集时无法产生梯度反馈的问题。MPDIoU 的定义公式(3-4)所示：

$$MPDIoU = \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \quad (3-4)$$

式中 A 、 B 分别为真实框和预测框， w 、 h 分别为边界框的宽和高， d_1 、 d_2 分别为真实框左上角到预测框左上角、真实框右下角到预测框右下角的距离，其中 d_1 、 d_2 定义为式(3-5)：

$$\begin{aligned} d_1 &= \sqrt{(x_1^B - x_1^A)^2 + (y_1^B - y_1^A)^2} \\ d_2 &= \sqrt{(x_2^B - x_2^A)^2 + (y_2^B - y_2^A)^2} \end{aligned} \quad (3-5)$$

式中 (x_1^A, y_1^A) 、 (x_2^A, y_2^A) 分别为真实框的左上角和右下角点的坐标， (x_1^B, y_1^B) 、 (x_2^B, y_2^B) 分别为预测框的左上角和右下角点的坐标。

$MPDIoU$ 作为损失函数，表示为式(3-6)：

$$L_{MPDIoU} = 1 - MPDIoU \quad (3-6)$$

3.3 交通标志检测实验与结果分析

3.3.1 训练环境和数据集

实验是在 Linux Ubuntu 18.04.6LTS 的操作系统上进行的, 所用到的硬件设备为 Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz 处理器, 2 块 24GB 显存的 NVIDIA RTX A5000 显卡, 使用 CUDA11.3 为 GPU 对图像处理进行加速。实验所用到的编程环境为 Python3.8, 使用 PyTorch1.12 框架, 超参数设置批大小(batch size)为 16, 训练周期(epochs)为 300, 初始学习率(learning rate)为 0.01。

在本实验中采用了由清华大学和腾讯联合整理并公开发布的 Tsinghua-Tencent 100K(TT100K)数据集。该数据集专注于中国交通标志的识别与检测任务, 并且数据集具有庞大的基数、丰富的标注信息以及高分辨率图像等优点。虽然数据集基数庞大, 包含丰富的交通标志类别, 但有一半以上的交通标志类别在数据集中出现的次数少于 50, 不能用来训练, 这种数据的不平衡性可能导致模型训练时出现过拟合。为了解决这个问题, 本文对数据集进行了筛选, 只留下了样本数超过 50 的 45 个类别。这样筛选后, 得到了 9170 张标注有交通标志的图片及其对应的标签文件。

3.3.2 实验性能评价指标

在评估模型性能时, 本文选用了计算量、准确率(Precision, P)、召回率(Recall, R)和平均精度(mean Average Precision, mAP)作为主要的衡量指标。

计算量反映了网络模型的计算复杂度, 以每秒执行 10 亿次浮点运算(GigaFLOPS, GFLOPS)为单位。计算量越大, 意味着需要更多的计算资源, 模型运算所消耗的时间也会相应增长, 其计算方法如公式(3-7)中所示:

$$\text{FLOPs} = 2 \times H \times W \times (C_{in} \times K^2 + 1) \times C_{out} \quad (3-7)$$

式中 H 和 W 分别表示卷积层的高度和宽度, C_{in} 和 C_{out} 分别表示输入和输出的通道数, K 表示卷积核的大小。

准确率 P 是指预测正确的正样本所占的比例, 它衡量了模型识别正样本的准确性, 其计算方法如公式(3-8)中所示:

$$P = \frac{TP}{TP + FP} \quad (3-8)$$

其中, TP 表示真正例(True Positive), 即预测为正且实际为正的样本数量; FP 表示假正例(False Positive), 即预测为正但实际为负的样本数量。

召回率 R 衡量了模型正确检测到的正样本与所有实际正样本之间的比例, 是对模型检测正样本能力的指标, 其计算方法如公式(3-9)中所示:

$$R = \frac{TP}{TP + FN} \quad (3-9)$$

其中, FN 表示假反例(False Negative), 即预测为负但实际为正的样本数量。

平均精度 mAP 与准确率 AP 值密切相关。在计算 mAP 时, 首先需要计算 $AP@0.5$, 它是对于每类具有 m 个正例的样本, 当交并比(IoU)阈值取 0.5 时, 每个正例对应的 P 值的平均值。 $mAP@0.5$ 是所有类别的 $AP@0.5$ 的平均值, 它反映了模型在准确率和召回率之间的权衡, 计算公式如式(3-10)所示:

$$\begin{aligned} AP@0.5 &= \frac{1}{m} \sum_{i=1}^m P_i \\ mAP@0.5 &= \frac{1}{n} \sum_{k=1}^n AP@0.5_k \end{aligned} \quad (3-10)$$

3.3.3 消融实验分析

为了全面评估四种优化手段的有效性, 在 TT100K 数据集上进行了详细的消融研究。构建了四个不同的模型版本, 并在每个版本中逐步引入了不同的改进策略。Model1 是在 YOLOv8s 的基础上增加了专门用于小目标检测的检测头。Model2 在 Model1 的基础上进一步强化了模型的特征提取能力, 在主干网络中加入了 CBAM 注意力模块。Model3 在 Model2 的基础上对原网络中的 C2f 模块进行了全面升级, 将所有的 Bottleneck 模块替换为了更为高效的 RepGhostBottleneck 模块。Model4 在 Model3 的基础上进一步优化了模型的训练过程, 通过将 CIOU 损失函数替换为 MPDIoU 损失函数, 使得在训练阶段模型能更加专注于提升边界框的准确性和定位性能。

表 3-1 TT100K 数据集上四种改进方法的消融研究

Model	小目标 检测头	CBAM	RepGhost	MPDIoU	Precision/%	Recall/%	mAP@0.5/%	GFLOPs
YOLOv8s	—	—	—	—	84.5	75.4	84.3	28.8
Model1	✓	—	—	—	85.6	78.3	86.7	38.3
Model2	✓	✓	—	—	87.7	79.4	87.9	38.5
Model3	✓	✓	✓	—	88.6	81.6	89.1	28.3
Model4	✓	✓	✓	✓	89.3	85.1	90.4	28.3

通过分析表 3-1 中的数据，当在 YOLOv8s 的基础上增加针对较小交通标志数据集的更大尺度检测头后，模型的 mAP@0.5 从 84.3% 提升至 86.7%。这一提升表明，通过保留更丰富的小目标特征信息，有效地减少了图像中的误检和漏检问题。通过引入 CBAM 注意力模块，Model2 相较于 Model1 在几乎不增加计算复杂度的前提下，mAP@0.5 提升了 1.2%。这一结果表明，CBAM 注意力机制成功增强了网络提取重要特征的能力，有效提升了对小目标的检测性能。Model3 在引入 RepGhostConv 模块后，相较于 Model2，计算复杂度降低了 26.5%，而 mAP@0.5 却上涨了 1.2%。这表明，在轻量化网络结构的同时，也可以实现精度的提升。这一改进在保持模型性能的同时，降低了计算资源的消耗。通过用 MPDIoU 损失函数替代 CIoU 损失函数，Model4 相较于 Model3 在检测精度上有所提高。这意味着新引入的损失函数更加高效地处理了边界框之间的重合问题，从而增强了模型的检测准确性。综上所述，与 YOLOv8s 相比，本文所提出的方法在检测精度方面实现了明显的进步，mAP@0.5、准确率和召回率分别提高了 6.1%、4.8%、9.7%，同时在计算量方面还实现了 1.7% 的下降。这些改进共同证明了本文算法在交通标志检测任务中的优越性。

从图 3-10 中可以看到在训练的前 30 轮中，Model4 模型表现出显著的进步。在这一阶段，损失函数值迅速降低，而准确度、召回率和 mAP@0.5 值则迅速提升。然而，随着训练轮数的增加，特别是在 100 轮后，这些指标的提升速度开始放缓。当轮数达到 260 轮后，损失曲线均趋于稳定，其他评估指标也达到了最优的结果。

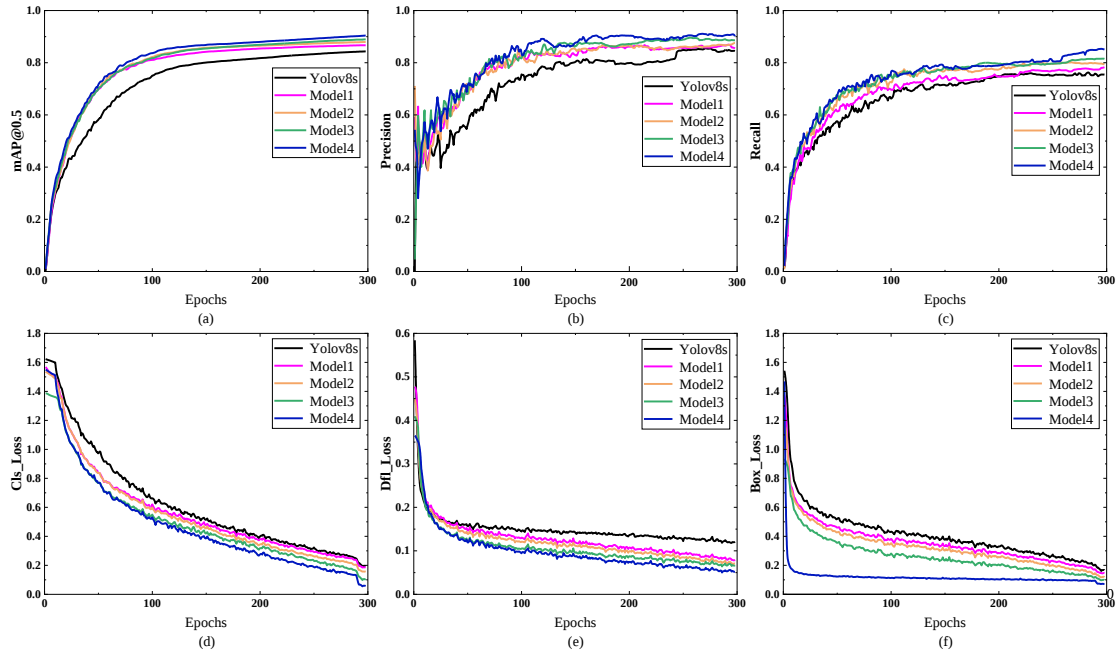


图 3-10 比较五个模型的训练过程。(a)mAP@0.5 曲线; (b)Precision 曲线; (c)Recall 曲线; (d)Cls_Loss 曲线; (e)Dfl_Loss 曲线; (f)Box_Loss 曲线

图 3-10(a)展示了 Model4 模型在训练过程中的 mAP 值变化。经过约 150 轮的训练, mAP 值达到了 88%, 并在此后保持稳定, 最高时甚至达到了 90.4%。这显示出模型在识别交通标志方面的高准确性和稳定性; 在图 3-10(b)中, 可以看到 Model4 的准确率随训练次数的增加而持续提升。当训练进行到 200 轮时, 准确率达到了 88.3%, 并继续提升至 89.3%。这进一步证明了模型在识别交通标志方面的有效性; 图 3-10(c)显示了 Model4 模型的召回率变化。在训练的前 50 轮中, 召回率迅速上升, 之后继续增长并最终达到了 85.1%。这表明模型在检测交通标志方面具有较高的召回率, 即能够准确地识别出大部分目标标志; 图 3-10(d)、(e)、(f)展现了 Model4 模型在不同训练阶段分类损失、动态冻结损失和边界框损失的变化, 当模型迭代 260 次时, 损失值基本稳定下来, 这表明网络已经基本收敛并达到了较好的性能; 相较于其他模型, Model4 在检测和识别交通标志方面的表现更加出色, 具有更快的收敛速度和更高的准确性, 进一步证实了其有效性和优势。总的来说, Model4 在交通标志的检测和识别任务上呈现了优异的性能, 满足甚至超过了预期的标准。

3.3.4 对比试验分析

对改进的 YOLOv8s 模型和一些具有代表性的检测模型在 TT100K 数据集上进行测试，训练结果如表 3-2 所示。这些结果验证了本文算法在交通标志检测方面的有效性，经过改进的 YOLOv8s 模型不仅提升了检测性能，而且实现了网络结构的轻量化，便于在边缘计算设备上部署。

表 3-2 不同网络模型检测性能对比

Model	Precision/%	Recall/%	mAP@0.5/%	GFLOPs
Faster RCNN	31.66	76.27	60.97	14.2
YOLOv3-tiny	73.4	72.1	73.8	9.2
YOLOv4	81.1	80.5	80.4	13.5
SSD300	42.8	65.7	51.6	15.4
YOLOv5s	83.3	78.2	83.3	16.1
YOLOv7	89.6	76.3	84.6	105.1
YOLOv8s	84.5	75.4	84.3	28.8
本文算法	89.3	85.1	90.4	28.3

根据表 3-2 的数据对比，本文提出的交通标志检测模型在各项指标上均表现优异。尽管 YOLOv7 模型在精确度上稍高，达到了 89.6%，但本文的算法也取得了 89.3% 的精确度，与 YOLOv7 相差无几。更为关键的是，本文算法在计算效率上大大优于 YOLOv7，实现了对性能与计算成本的有效平衡。特别值得一提的是，本文算法在召回率上取得了显著的优势，达到了 85.1%，这表明它能更全面地识别出实际的交通标志，减少了漏检的可能性。此外，本文算法的平均精确度均值 (mAP@0.5) 以 90.4% 的表现位列第一，这证明了在各种不同难度的检测任务中，本文算法都能够保持较高的稳定性和准确性。尽管本文算法的计算复杂度为 28.3GFLOPs，相较于 Faster RCNN、YOLOv3-tiny、YOLOv4、SSD300 和 YOLOv5s 等模型有所增加，但考虑到其在精确度、召回率和 mAP@0.5 等方面的显著提升，这种增加的计算复杂度是完全值得的。

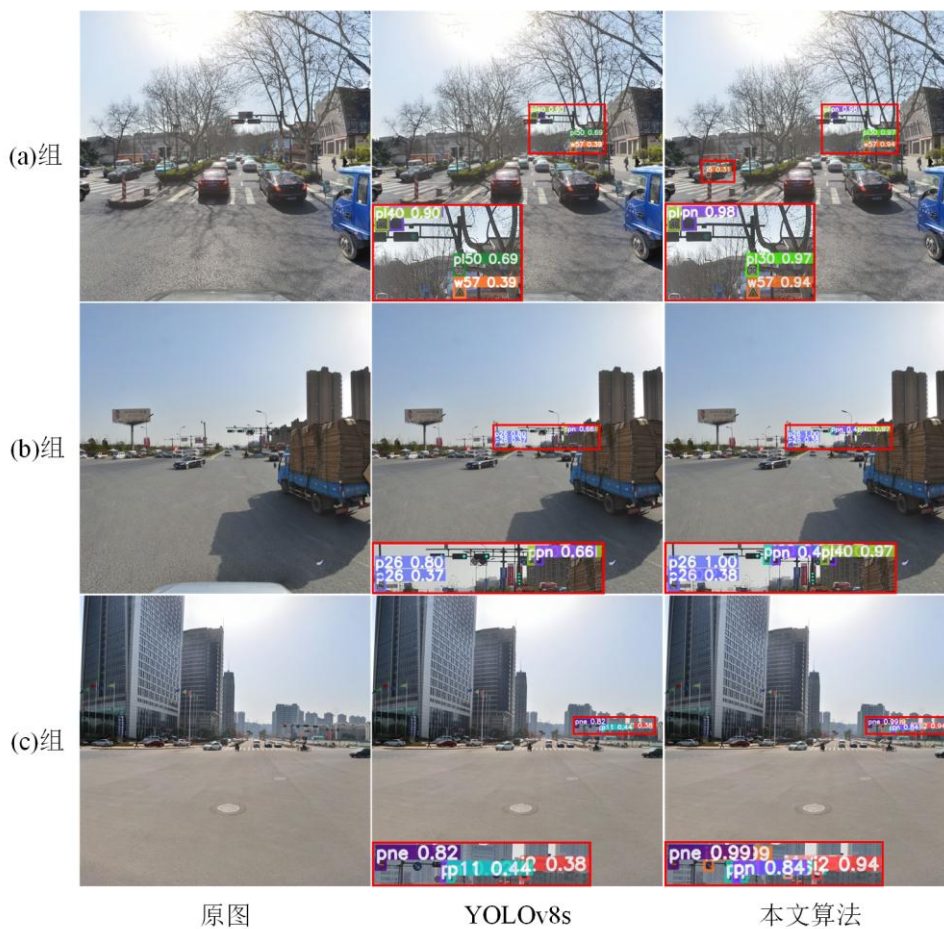


图 3-11 改进前后检测结果

通过分析 YOLOv8s 与本文算法在 TT100K 数据集上处理交通标志的结果，可以从可视化图片中观察到显著的差异。如图 3-11 所示，首先，观察(a)组实验结果，可以清楚地看到本文算法在复杂环境中对孤立小型交通标志的识别更加准确，这充分证明了本文算法在解决交通标志漏检、误检问题上的有效性，特别是在复杂的道路条件下。进一步的通过对比(b)组和(c)组的实验结果，本文算法在处理密集和小尺寸交通标志时也表现出色，不仅检测效果更佳，而且识别精度也显著提高。这充分证实了本文算法在识别远处小型交通标志方面的改进。

3.4 本章小结

本章介绍了基于 YOLOv8s 的交通标志检测方法及其改进。首先，介绍了 YOLOv8s 模型的结构和工作原理，并分析了该模型在交通标志检测中的应用。随后，本章提出了一些针对 YOLOv8s 模型的改进措施，包括结合轻量化设计的

Ghost 模块、引入 **CBAM** 注意力机制，以及针对小目标优化的检测层。这些改进有效提升了模型在复杂环境中的检测精度和效率。通过在 **TT100K** 数据集上的实验验证，改进后的模型在各项性能指标上表现优异，特别是在检测精度、召回率和 **mAP** 值方面取得了显著的提高。本章为后续路径规划系统的实现提供了坚实的目标检测基础。

第4章 基于改进蚁群混合 AAPF 的路径规划算法

第三章提出了一种基于改进 YOLOv8s 的高效检测模型，有效提升了交通标志识别的准确性。然而，仅依赖精准的交通标志检测仍难以满足复杂交通环境下的路径规划需求。因此，在已有研究的基础上，本章提出了一种改进蚁群算法混合人工势场法的路径规划方法，引导无人车在复杂道路条件下实现自主导航。

4.1 人工势场法

人工势场法(Artificial Potential Field, APF)由 Khatib 提出^[62]，是一种基于力的路径规划算法，广泛应用于机器人和自动驾驶领域，该算法通过将智能车与目标点之间的距离视为吸引势能，同时引入障碍物的斥力，通过计算合力来确定智能车的运动方向和速度。智能车会在引力和斥力的作用下，逐步朝目标移动，直到到达目标或无法继续为止。

尽管人工势场法简单易懂、实现方便、计算速度快，但其局限性较为明显。首先，算法在路径规划时无法克服局部最优，尤其在复杂环境中；其次，参数设置对结果影响较大，不恰当的参数设置可能导致无人车无法有效避障或生成的路径并非最优。此外，算法在处理动态障碍物时也存在困难。因此，该方法仅适用于简单场景，但在复杂环境下常需通过优化和改进以提高适应性^[63]。

4.1.1 传统人工势场法

人工势场法是一种基于势场理论的智能车路径规划方法^[64]，旨在通过引力与斥力的相互作用来引导智能车完成从起始点到目标点的导航。在此方法中，智能车被视作一个受力物体，其运动轨迹由目标点和环境中障碍物产生的力所决定。通过构建势能函数，能够精确描述智能车与目标点及障碍物之间的相互作用力，从而推导出其运动轨迹。

目标点通过引力作用于智能车，促使智能车朝向目标点移动；而障碍物则通过斥力作用于智能车，防止其与障碍物发生碰撞。在引力和斥力的合力作用下，智能车根据实时环境变化动态调整其运动方向和速度，确保路径规划的优化性与

避障效果。通过这种力的综合作用，智能车能够高效、安全地规划路径并最终到达目标点。该方法在动态环境中表现出较强的适应性，能够实时更新路径规划，确保智能车在不断变化的环境中持续有效地执行任务。

在人工势场法中，智能车的受力分析如图 4-1 所示。

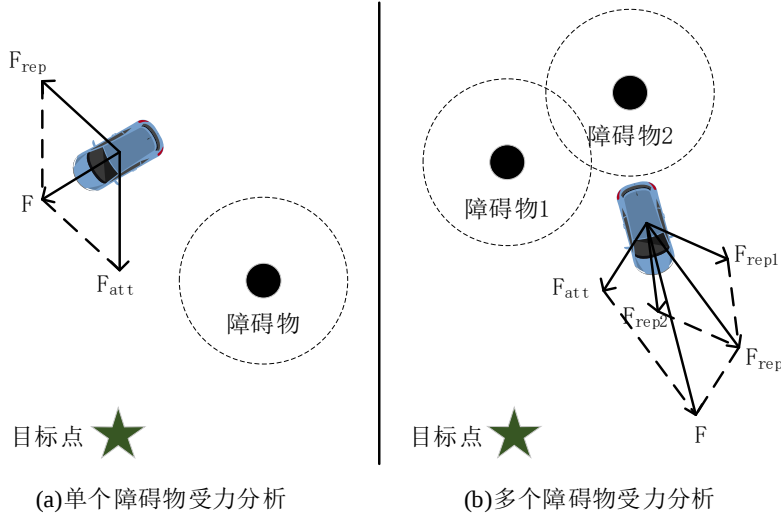


图 4-1 人工势场受力分析图

人工势场法的引力势场函数如式(4-1)所示：

$$U_{att}(q) = \frac{1}{2} \xi \rho^2(q, q_g) \quad (4-1)$$

$$\rho(q, q_g) = \sqrt{(x - x_g)^2 + (y - y_g)^2} \quad (4-2)$$

式中 U_{att} 表示由目标点产生的引力场， ξ 代表引力场强度系数， $\rho(q, q_g)$ 为智能车到目标点的欧氏距离。

目标点产生的引力是引力势场的负梯度，力的方向由智能车指向目标点。

$$F_{att}(q) = \xi \rho(q, q_g) \quad (4-3)$$

人工势场法的斥力势场函数如式(4-4)所示：

$$U_{rep}(q) = \begin{cases} \frac{1}{2} \eta \left[\frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho_0} \right]^2, & 0 \leq \rho(q, q_{obs}) \leq \rho_0 \\ 0, & \rho(q, q_{obs}) \geq \rho_0 \end{cases} \quad (4-4)$$

式中 U_{rep} 表示由障碍物产生的斥力场， η 代表斥力场强度系数， $\rho(q, q_{obs})$ 是智能车当前位置和障碍物之间的欧氏距离， ρ_0 为障碍物的斥力影响半径。当智能车与

障碍物的距离大于 ρ_0 时，斥力场为零，智能车不受其影响；而当距离小于 ρ_0 时，智能车将受到斥力场的影响，斥力的大小可通过式(4-5)计算。

$$F_{rep}(q) = \begin{cases} \eta \left[\frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho_0} \right] \frac{1}{\rho^2(q, q_{obs})} & , 0 \leq \rho(q, q_{obs}) \leq \rho_0 \\ 0 & , \rho(q, q_{obs}) \geq \rho_0 \end{cases} \quad (4-5)$$

智能车在运动过程中，不仅受到目标点引力的影响，还会受到多个障碍物斥力的共同作用，最终的合力可用 $F(q)$ 表示， m 为对智能车起作用的障碍物的数量。

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (4-6)$$

$$F(q) = F_{att}(q) + \sum_{i=1}^m F_{rep}(q) \quad (4-7)$$

在障碍物较少的场景中，人工势场法能够有效实现路径规划并找到最优路径；然而，在障碍物密集的环境下，传统的人工势场法会暴露出一些问题，尤其是在多障碍物和动态变化的场景下，常常会出现目标不可达或陷入局部极小点的情况，导致无人车无法顺利完成任务或路径规划效果不佳。

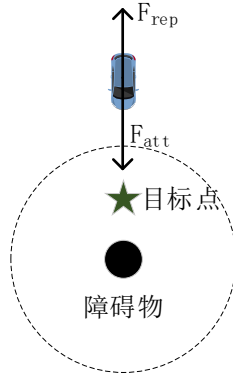


图 4-2 目标不可达示意图

如图 4-2 所示为智能车不能到达目的地的示意图，当智能车从起点出发逐步接近终点时，如果目标点位于障碍物附近，车辆会同时受到目标点的吸引力和障碍物的斥力。然而，当车离目标点较近时，吸引力急剧减弱，而斥力依然较大，导致合力指向远离目标的方向，从而使车辆无法到达终点。

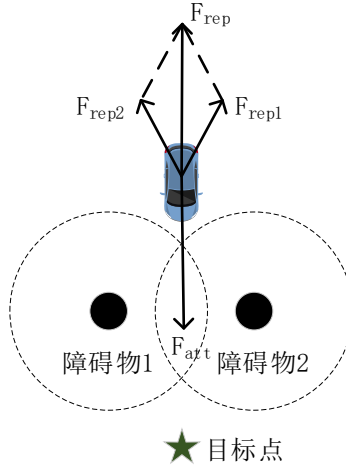


图 4-3 局部极小点示意图

如图 4-3 所示，在路径搜索过程中，智能车受到多个力的作用，这些力相互作用形成合力，驱动智能车沿合力方向前进，但在局部极小点，合力为零，智能车停滞。局部极小点通常出现在智能车被多个障碍物困住时，障碍物产生的斥力与目标引力相互抵消，使得智能车停滞不前，无法到达目标点。

4.1.2 改进的人工势场法

为克服传统人工势场法在路径规划中的局限性，本文根据智能车与目标点的距离动态调整斥力势场，自适应地调整排斥力，从而避免无人车陷入局部最优解。优化后的斥力势场函数如式(4-8)所示：

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta \left[\frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho_0} \right]^2 \rho^3(q, q_g) & , 0 \leq \rho(q, q_{obs}) \leq \rho_0 \\ 0 & , \rho(q, q_{obs}) \geq \rho_0 \end{cases} \quad (4-8)$$

式中 $\rho(q, q_g)$ 表示无人车当前位置与目标点之间的欧氏距离。障碍物产生的斥力势场的大小与 $\rho(q, q_g)$ 成正比。当智能车逐渐接近终点时，障碍物的斥力势场逐渐减小，当到达终点后，势场的总和变为零。

斥力大小可通过式(4-9)计算：

$$F_{rep}(q) = \begin{cases} F_{rep1}(q) + F_{rep2}(q) & , 0 \leq \rho(q, q_{obs}) \leq \rho_0 \\ 0 & , \rho(q, q_{obs}) \geq \rho_0 \end{cases} \quad (4-9)$$

$$F_{rep1}(q) = \eta \left[\frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho_0} \right] \frac{1}{\rho^2(q, q_{obs})} \rho^3(q, q_g) \quad (4-10)$$

$$F_{rep2}(q) = \frac{3}{2} \eta \left[\frac{1}{\rho(q, q_{obs})} - \frac{1}{\rho_0} \right]^2 \rho^2(q, q_g) \quad (4-11)$$

在该式中，分量 $F_{rep1}(q)$ 是对式(4-8)中的 $\rho(q, q_{obs})$ 求偏导数得到的，其方向由障碍物指向智能车；而分量 $F_{rep2}(q)$ 是对 $\rho(q, q_g)$ 求偏导数的结果，方向由智能车指向目标点。当终点在障碍物附近时，改进的人工势场相当于给无人车施加了一个指向目标的力，这一额外的引力相当于增加目标引力的作用，克服斥力过大带来的不利影响，确保无人车朝目标移动。改进后的受力分析示意图如图 4-4 所示。

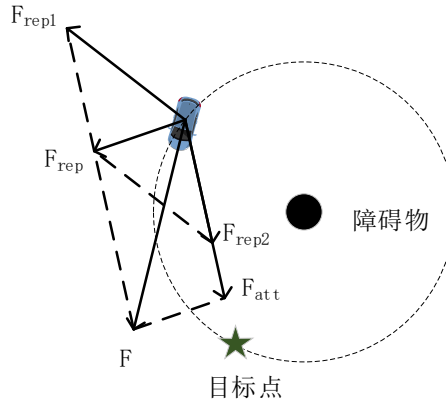


图 4-4 改进人工势场受力分析图

4.2 蚁群算法

蚁群算法(Ant Colony Optimization, ACO)是受蚂蚁寻找食物启发的一种群体智能优化算法，其核心思想在于模拟蚂蚁利用信息素进行交流，以协同寻找最优路径。在行进过程中，蚂蚁会根据信息素浓度选择行进路线，浓度较高的路径更容易吸引蚂蚁，而信息素随时间蒸发，这一机制推动了路径的不断优化，最终使搜索过程收敛至最优解。

由于蚁群算法的全局搜索能力，它被广泛应用于复杂的路径规划问题中。该算法能有效应对复杂环境中的路径搜索，但在实际应用中，由于其较慢的收敛速

度，容易陷入局部最优解。尤其是在遇到像 U 型锁这样的路径陷阱时。算法可能无法找到最优路径，甚至出现死锁现象。因此，需要对蚁群算法的初始参数进行合理设置，以提高算法的性能和避免死锁问题。

4.2.1 蚁群算法相关参数

在蚁群算法中，关键的参数包括蚁群规模 M 、信息素启发因子 α 、期望启发因子 β 、信息素挥发因子 ρ 、信息素浓度 Q 以及迭代次数 $iter$ 。每个参数都在不同方面对算法的性能产生作用，具体分析如下：

(1) 蚁群规模 M

蚁群规模就是表示参与搜索的蚂蚁数量。蚂蚁的数量影响算法的搜索能力和计算效率。过多的蚂蚁可能导致计算量过大，过少则可能导致搜索不充分。

(2) 信息素启发因子 α

信息素启发因子 α 控制着蚂蚁在选择路径时对信息素的敏感度，决定了信息素在路径选择中的重要性。当 α 值较大时，蚂蚁会更加倾向于选择信息素浓度较高的路径，即更容易沿着之前蚂蚁走过的路径前进。这样，信息素较强的路径会吸引更多蚂蚁，加速这些路径的强化。

(3) 期望启发因子 β

期望启发因子 β 决定了蚂蚁在选择路径时对启发函数信息的依赖程度，影响算法的全局搜索效率。启发函数通常用来衡量路径的吸引力，如路径的距离或时间等。当 β 值较大时，蚂蚁会更多地依赖启发函数的提示，倾向于选择那些在启发函数上具有优势的路径，加快寻找潜在的最优路径。

(4) 信息素挥发因子 ρ

信息素挥发因子 ρ 控制着路径上信息素浓度的衰减速度， ρ 值越大，路径上信息素浓度蒸发越快，此时找到最优路径也就越快，但如果 ρ 值过大，会使得路径的搜索能力变得很低，导致算法无法准确找到全局最优路径。

(5) 信息素浓度 Q

信息素浓度指的是路径上吸引蚂蚁选择该路径的信息强度。若某条路径信息素浓度较高，则蚂蚁选择该路径的概率会显著增加，这通常意味着该路径在局部范围内表现出较优的特性。

(6) 迭代次数 iter

蚁群算法中的迭代次数决定了算法执行的循环次数,对求解结果和效率都有重要影响。如果迭代次数设置过少,算法可能在找到最优路径之前就结束,导致结果不够理想;而如果迭代次数过多,虽然可能提高解的精度,但会降低算法效率,增加计算负担,实时性也会受到影响,甚至出现多次迭代得到相同路径的情况,从而浪费计算资源。

4.2.2 蚁群算法基本原理

蚁群算法源自对蚂蚁寻找食物行为的模拟。在寻找食物的过程中,蚂蚁会在不同路径上留下各异浓度的信息素来反映路径的优劣,其他蚂蚁则根据信息素浓度来判断选择哪条路径,浓度较高的路径被选中的概率也更大。随着优质路径的信息素不断累积,蚂蚁逐步趋向最优路径。

蚂蚁的移动方向受路径上信息素浓度影响。在第 t 次迭代中,蚂蚁 k 从节点 i 前往节点 j 的概率可由式(4-12)计算。

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in allowed} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta} & , j \in allowed \\ 0 & , otherwise \end{cases} \quad (4-12)$$

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (4-13)$$

式中 α 是信息素启发因子, β 是期望启发因子, τ_{ij} 是路径 (i, j) 上的信息素浓度,

η_{ij} 是路径的启发信息,通常与路径的质量相关, d_{ij} 表示路径 (i, j) 的欧氏距离。

allowed 是当前节点 i 可选择的所有邻接节点集合。

在蚁群算法中,信息素更新是一个关键过程,它通过挥发和蚂蚁留下的信息素共同作用来引导搜索。本文采用蚁周模型来更新信息素。当所有蚂蚁完成一次迭代后,路径上的信息素浓度将进行更新,如式(4-14)所示。

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (4-14)$$

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (4-15)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & , \text{蚂蚁}k\text{从节点}i\text{到节点}j \\ 0 & , \text{其他} \end{cases} \quad (4-16)$$

式中 ρ 是信息素挥发系数。 L_k 是在本次迭代中蚂蚁 k 走过的路径总长度。 Q 是信息素强度常数，用来控制单个蚂蚁对信息素的贡献量。 $\Delta\tau_{ij}$ 是路径上信息素浓度的变化量。

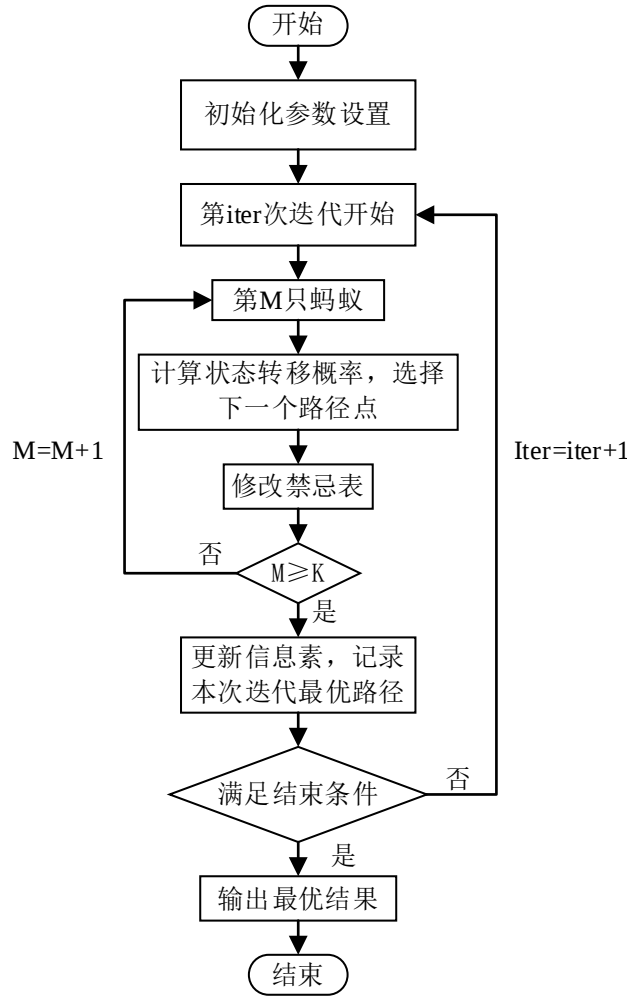


图 4-5 蚁群算法流程图

蚁群算法流程图如图 4-5 所示，这个流程图详细描述了蚁群算法的执行过程。首先，算法初始化参数并开始执行第 $iter$ 次迭代。接着，进入每次迭代中的循环，首先计算并选择蚂蚁的下一步路径。每只蚂蚁根据路径上的信息素浓度和启发信息，选择下一个路径点，然后更新其禁忌表，避免重复选择已经走过的路径。随着蚂蚁的选择完成，蚂蚁编号 M 递增，表示新的蚂蚁开始路径选择。完成所有

蚂蚁的路径选择后，检查是否达到指定的蚂蚁数量，如果是，则进入信息素更新阶段并记录当前迭代中最优路径。之后，算法会增加迭代次数 *Iter*，并判断是否满足终止条件。如果满足条件（如最大迭代次数或达到期望解），则输出最优结果，表示算法结束。如果不满足终止条件，则继续进入下一次迭代。最终，经过多轮迭代，蚁群算法会输出最优路径作为结果。

4.3 改进的蚁群算法

近年来，蚁群算法在路径规划中得到了广泛应用，但也存在一些局限性，需要进一步改进^[65]。Liu 等人^[66]提出了一种结合信息素扩散和几何局部优化的改进蚁群优化算法(ACO-PDG algorithm)，该算法综合利用了信息素扩散机制和几何局部优化方法的优势，信息素扩散机制引导蚂蚁搜索更高适应度子空间，几何局部优化方法则用于优化路径。Dai 等人^[67]提出了一种改进的蚁群算法，该算法综合利用了 A*算法和最大-最小蚂蚁系统的优点，引入 A*算法的评估函数和弯曲抑制算子来改进蚁群算法的启发式信息，借助最大-最小蚂蚁系统用于优化信息素更新及避免过早收敛。

传统的蚁群算法在路径规划任务中，由于初始信息素的均匀分布，导致搜索过程缺乏指导性，容易陷入局部最优，且在迭代后期难以快速收敛到最优解。为此，本文提出了一种基于改进蚁群混合 AAPF 的路径规划算法。该算法构建了非均匀分配的初始信息素设置，减少初始搜索的盲目性，能够有效引导蚂蚁快速朝目标前进，减少搜索时间并加速算法收敛；同时，在蚁群搜索过程中引入自适应信息素更新策略，根据路径质量动态调整信息素浓度，提高全局搜索能力；结合改进的人工势场法，通过引力与斥力相结合的势场调控机制，引导蚁群避障并优化路径平滑性。此外，为防止路径规划陷入局部最优状态，本文引入了惩罚系数，从而提升了算法跳出局部最优解的能力。

4.3.1 初始化信息素分布

蚁群算法在路径规划初期，由于地图上所有节点的信息素浓度相同，蚁群算法中的蚂蚁会随机朝各个方向移动。虽然这种随机搜索策略有助于发现多样化的

路径，但同时也导致收敛速度缓慢，特别是在复杂环境中，蚂蚁可能会在较远路径上徘徊，从而浪费计算资源并延长搜索时间。

为了加快算法的收敛速度，本文通过当前节点、下一个节点与起止点连线之间的相对距离来计算初始信息素，如式(4-17)所示：

$$\tau_{ij}(0) = \frac{d_{SG}}{d_{Si} + d_{ij} + d_{jG}}, j \in C \quad (4-17)$$

式中 $\tau_{ij}(0)$ 是蚁群初始时刻的信息素浓度， d_{SG} 是起点到终点的欧几里得距离， d_{Si} 是起点到当前节点的欧几里得距离， d_{ij} 是当前节点与下一个节点之间的欧几里得距离； d_{jG} 是下一个节点到终点的欧几里得距离； C 是下一个节点的集合；式中 $d_{Si} + d_{ij} + d_{jG}$ 的值越小，路径上的初始信息素越大；反之则越小。相比传统的均匀分布信息素策略，这种根据位置关系设置不均匀分布的初始信息素，不仅加快了搜索过程，还提高了全局搜索能力，从而使蚂蚁在更短时间内找到最优路径。

4.3.2 自适应信息素浓度

在传统蚁群算法中，信息素浓度的更新通常是静态的，即在每一次迭代后，所有路径的信息素浓度按固定比例增加或蒸发。这种方法虽简便有效，但随着迭代的进行，次优路径的信息素占比增大，导致种群多样性下降，搜索效率降低，收敛变慢。为解决这一问题，本文采用蚁周模型更新信息素，依据上次迭代最优路径的表现调整信息素，以减少质量较差路径对后续迭代的负面影响。同时，提出自适应信息素调整策略，动态优化信息素浓度，强化优秀路径的浓度，削弱次优路径，从而加速收敛并提升全局搜索能力。

$$\begin{cases} \tau_{ij}(t+1) = [1 - \rho(t)] \tau_{ij}(t) + \Delta \tau_{ij}(t, t+1) \\ \rho(t) = \rho(0) \cdot \left(1 - \frac{Iter(t)}{MaxIter} \right) \\ \Delta \tau_{ij}(t, t+1) = \sum_{k=1}^m \frac{Q}{L_k} \cdot f(Quality) \cdot \delta_{ij} \\ f(Quality) = w_1 \cdot \frac{1}{L} + w_2 \cdot S + w_3 \cdot D \end{cases} \quad (4-18)$$

在自适应信息素浓度调整中,信息素浓度的更新公式进行了改进,如式(4-18)所示,式中 $\rho(t)$ 是动态调整的挥发率, $\rho(0)$ 是初始挥发系数, $Iter(t)$ 是当前的迭代次数, $MaxIter$ 是最大迭代次数。随着迭代的进行,挥发系数逐渐减小,这样可以避免过早的收敛,也能适应不同搜索阶段的需求; δ_{ij} 是指示函数,表示蚂蚁是否经过该路径,当蚂蚁 k 走过该路径时, $\delta_{ij} = 1$, 否则为 0; $f(Quality)$ 是路径质量的加权因子, w_1 、 w_2 、 w_3 是对应的权重系数,用于调节每个因素对路径质量的影响程度, L 表示路径长度,路径越短,倒数值越大,表示路径质量越好, S 和 D 分别表示路径的平滑度和避障程度,值越大表示路径质量越好。

通过自适应信息素浓度调整策略,算法不仅在信息素更新方面实现了更高的灵活性,还能够根据路径的质量和搜索进程动态调整信息素的挥发率和增量,显著提高了路径规划的效率、质量和收敛速度。

4.3.3 惩罚系数

在蚁群算法中,局部最优解的出现通常是因为某些路径的信息素积累过多,导致蚂蚁对这些路径的选择倾向过高,最终形成局部最优解。为了解决这个问题,需要引入惩罚系数来降低已经过度搜索的路径的吸引力,从而促进全局搜索。

为了有效引入惩罚系数,需要有一种机制来识别局部最优路径。本文通过以下方式来度量路径的质量,识别出那些过度被选中的路径:

(1) 路径质量度量: 通过多次迭代后,对每一条路径的平均代价进行计算。如果某条路径的代价(路径长度或其他评估指标)持续处于较差状态,或者该路径被重复选择的频率过高,认为这条路径可能会导致局部最优解。路径质量可以采用路径长度的倒数进行量化,因为路径越长、质量越差时,惩罚应越大。

(2) 路径选择频率: 计算每条路径被选择的次数,如果某条路径的选择次数在多代中偏高,说明这条路径过度吸引了蚂蚁,可能导致搜索陷入局部最优。

当算法运行到第 K 代时,如果当前最优路径 L 未发生显著变化,则在下一轮信息素更新时,通过衰减最优路径上的信息素浓度来减少其影响。这一信息素衰减过程可以通过式(4-19)进行描述:

$$\gamma_{ij} = -u_1 \frac{L_{ij}}{L_{\max}} - u_2 \frac{N_{ij}}{N_{\max}} \quad (4-19)$$

式中 u_1 、 u_2 是调节参数，控制路径质量和选择频率对惩罚的影响。 L 是最优路径的长度， N 是最优路径在迭代中被选择的频率。

4.4 混合算法实现流程

随着路径规划需求的日益复杂，单一算法已无法满足多样化的实际应用需求。因此，探索综合性强、效率高的算法解决方案显得尤为重要。算法融合通过结合不同算法的优势，弥补各自的不足，拓展了应用范围。蚁群算法以其强大的全局搜索能力在路径规划中表现优异，而人工势场法则在局部路径生成中能够有效提供平滑、安全的路线。将这两种算法结合，既能发挥各自的优点，又能克服单一算法的缺陷，具有显著的应用价值。基于这一思路，本文提出了一种基于改进蚁群混合 AAPF 的路径规划算法，算法的具体实现步骤如下：

步骤 1：在 MATLAB 软件上使用栅格法建立二维地图环境。环境中的每个单元格长度设置为 1 米，黑色单元格表示障碍物，白色单元格表示可通行区域。设置起始点坐标和目标点坐标。

步骤 2：设置非均匀分配的初始信息素，初始化蚁群算法参数，包括蚁群数量、信息素浓度、启发函数权重等。

步骤 3：结合障碍物的排斥力与目标点的吸引力，使用改进的人工势场法计算势场力，每只蚂蚁根据状态转移概率公式选择下一节点。

步骤 4：判断当前路径是否可以继续前进，若没有可行路径，返回上一节点重新选择路径，同时将已走过的路径加入禁忌表。

步骤 5：判断所有蚂蚁是否完成路径规划。如果完成，保存当前最优路径。如果未完成，回到步骤 3 继续路径选择。

步骤 6：根据路径质量动态更新路径上的信息素浓度。信息素更新遵循自适应策略，同时更新势场力和启发函数。

步骤 7：在当前最优路径附近引入惩罚系数，对信息素进行动态调整。

步骤 8：清空禁忌表，增加迭代次数 $N=N+1$ ，继续进行路径选择与信息素更

新。

步骤 9: 判断算法是否达到最大迭代次数或路径质量是否达到预设标准。若达到终止条件, 停止迭代输出最优路径。如果未满足终止条件, 则继续迭代, 返回步骤 3。

算法的流程框图如图 4-6 所示:

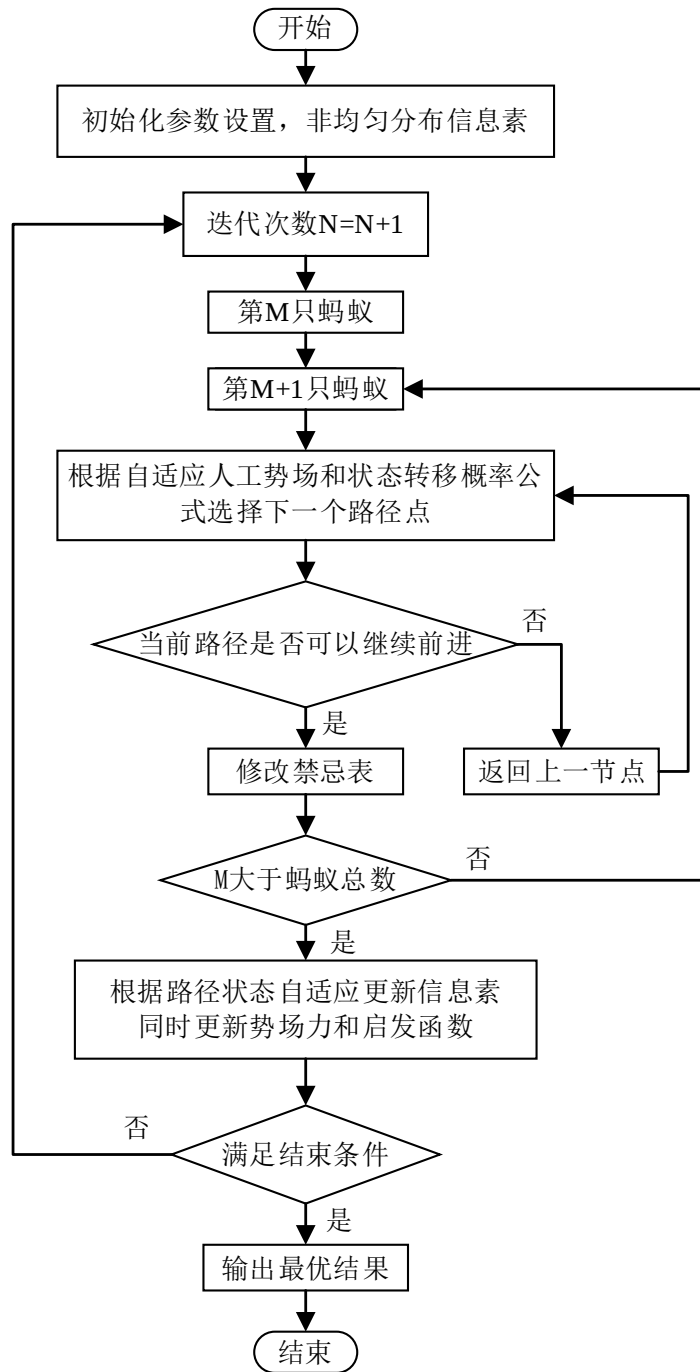


图 4-6 改进算法流程图

4.5 仿真实验分析

本节通过栅格法构建环境地图,实验在一台配置为 16GB 内存、2.30GHz CPU 的计算机上进行,操作系统为 Windows 11,使用 MATLAB 2022a 进行路径规划仿真。首先,采用栅格法创建 20×20 米的二维地图,设置不同位置的障碍物,起始点坐标为(0.5,19.5),目标点坐标为(19.5,0.5),白色方格表示可通行区域,黑色方格表示障碍物。实验中分别设置了较少、中等、较多障碍物占比的环境,测试算法在不同障碍物分布下的路径规划表现。

实验中改进算法的参数如表 4-1 所示:

表 4-1 改进算法参数取值

参数名称	参数取值
蚂蚁数量 M	50
信息素挥发系数 ρ	0.3
信息素启发因子 α	3
期望启发因子 β	4
最大迭代次数 $N_{\max}$	100
初始信息素浓度 τ_0	0.4
信息素更新强度 Q	7
信息素更新权重 w_1, w_2, w_3	0.4, 0.4, 0.2
惩罚调节系数 u_1, u_2	0.6, 0.4

4.5.1 少量障碍物环境中仿真结果

在少量障碍物占比的情况下,图 4-7 显示出不同算法的路径规划效果。传统蚁群算法的搜索效率较低,路径呈现出较多不规则震荡,并且容易陷入局部最优,主要由于初期信息素分布均匀,缺乏有效指引,导致搜索过程盲目且路径不稳定。相比之下,本文改进算法通过在目标点附近赋予较高的信息素浓度,提升了收敛速度,并结合势场力和局部搜索机制,使得路径更加平滑,避障更加合理。粒子群算法的路径相对平滑,但在局部规划上存在绕行距离较长的问题,不能规划出最优路径。总体而言,本文算法在路径平滑性、避障能力和局部优化方面均表现

优异，既缩短了路径长度，又降低了路径弯曲度，提高了整体规划效率。

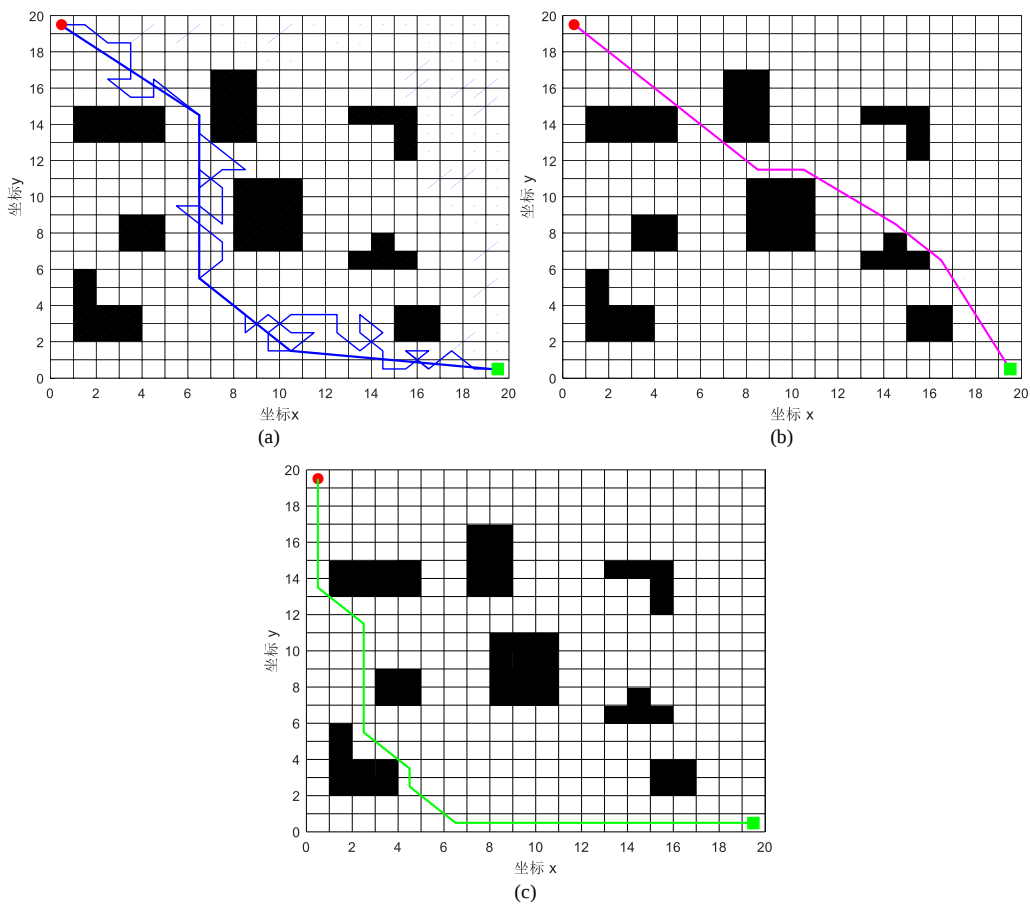


图 4-7 仿真实验结果 (a)传统蚁群算法 (b)本文算法 (c)粒子群算法

表 4-2 算法性能参数对比

算法	路径长度/m	运行时间/s
粒子群算法	34.485	3.296
传统蚁群算法	31.523	3.457
本文算法	27.851	2.968

为了减少实验的偶然性，每种算法均运行 10 次并取平均值，结果如表 4-2 所示。从表中的数据可以看出，本文改进算法在路径优化和计算效率方面均优于传统方法。本文算法生成的路径比传统蚁群算法缩短 3.672 米，相较于粒子群算法减少 6.634 米，有效降低了路径弯曲度，减少了不必要的绕行，从而获得更优的路径方案。此外，在计算效率方面，改进算法的运行时间比传统蚁群算法降低 14.1%，比粒子群算法降低 9.9%，显著提升了求解速度，加快了收敛过程。

4.5.2 中等数量障碍物环境中仿真结果

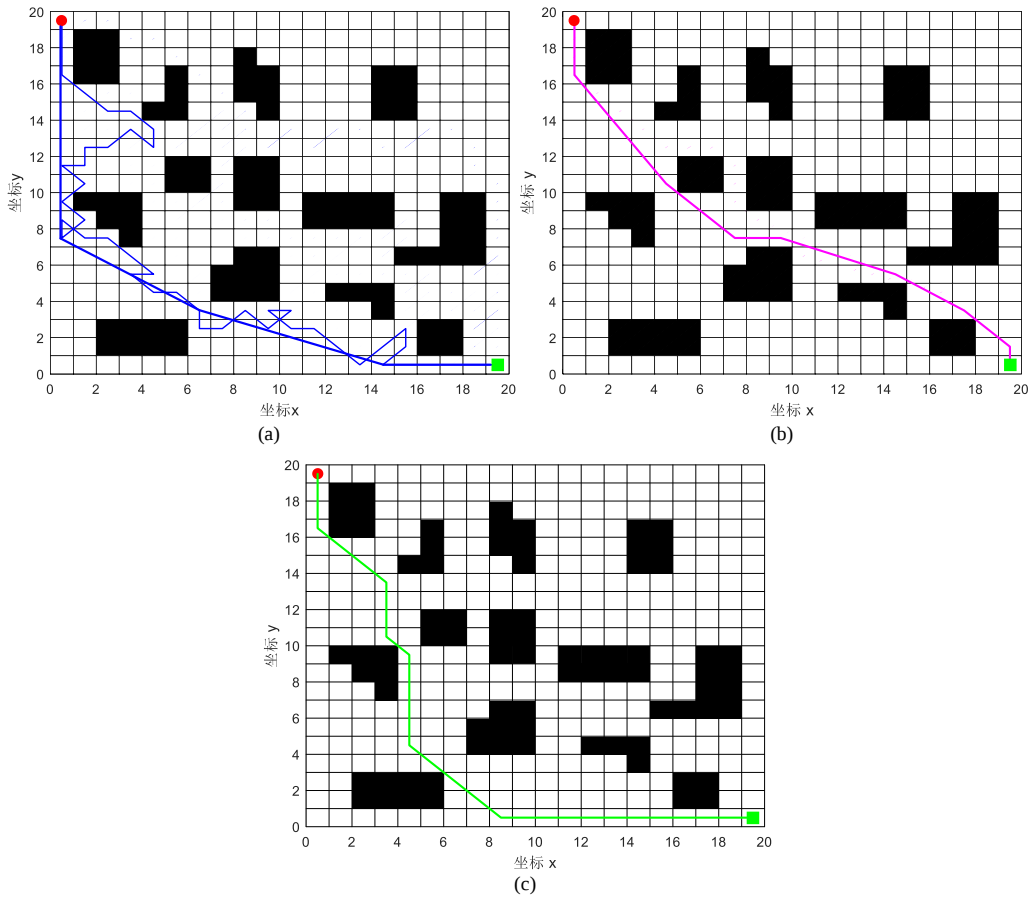


图 4-8 中等数量障碍物仿真实验结果 (a)传统蚁群算法 (b)本文算法 (c)粒子群算法

在中等数量障碍物占比的环境下，图 4-8 直观展示了不同算法的路径规划效果。传统蚁群算法生成的路径较长，拐点密集，且存在明显的震荡现象，主要由于信息素更新机制较弱，导致搜索易受次优路径干扰，影响收敛效率。相比之下，本文改进算法通过优化信息素更新策略，引入自适应状态转移规则和惩罚机制，使得路径更加平滑，避障更加高效，减少了不必要的绕行，提高了全局收敛能力。粒子群算法虽然路径较平滑，但局部规划时存在较长绕行的情况，影响路径整体优化效果。综合来看，改进算法能够快速避开次优解的干扰，减少路径的冗长和不必要的转折，从而更高效地找到全局最优解，快速完成路径规划任务。

为减少实验的偶然性，每种算法均运行 10 次并取平均值，表 4-3 的数据表明，改进算法在路径规划和计算效率方面均表现优异。与传统蚁群算法相比，改进算法的平均路径长度缩短 3.384 米，相较于粒子群算法则减少 4.241 米，充分

体现了其在寻优能力上的显著优势。此外，改进算法的运行时间较传统蚁群算法减少 18.3%，比粒子群算法降低 16.9%，进一步证明了其更高的计算效率与更强的收敛性能，使路径规划更高效。

表 4-3 算法性能参数对比

算法	路径长度/m	运行时间/s
粒子群算法	33.314	4.608
传统蚁群算法	32.457	4.682
本文算法	29.073	3.825

4.5.3 较多数量障碍物环境中仿真结果

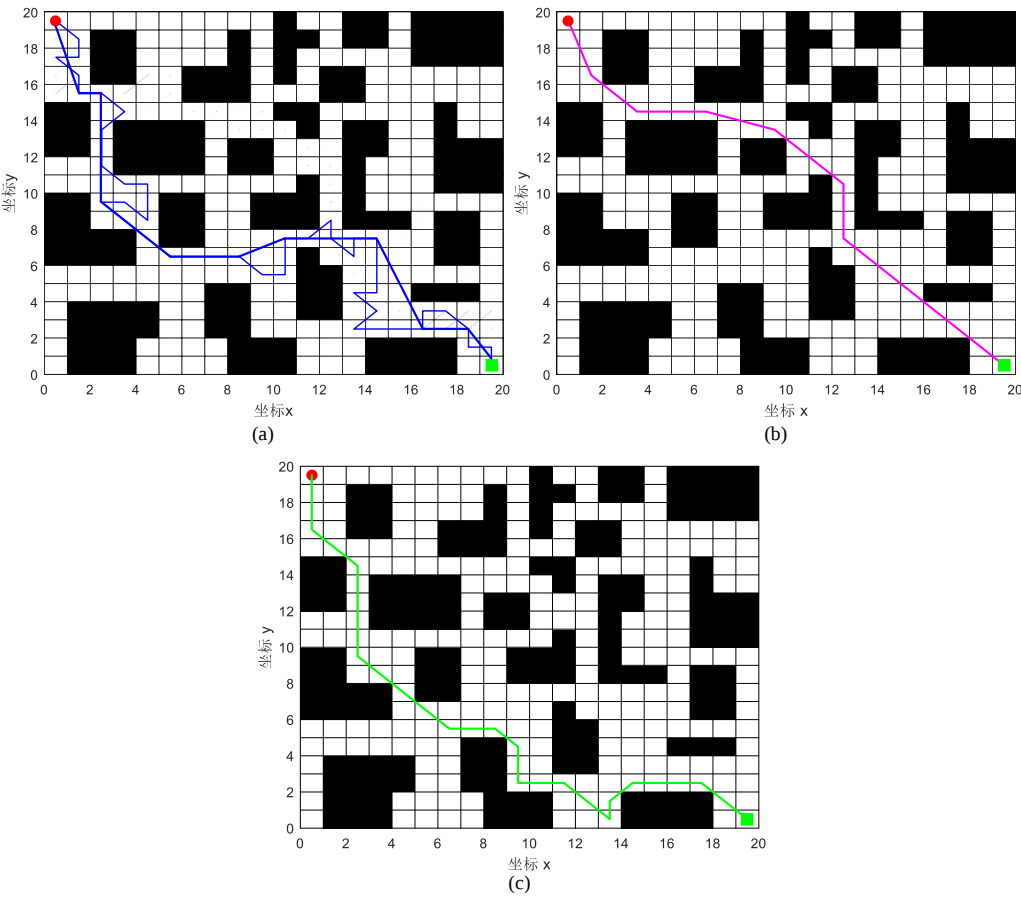


图 4-9 较多数量障碍物仿真实验结果 (a)传统蚁群算法 (b)本文算法 (c)粒子群算法

根据图 4-9 的路径规划对比分析，在障碍物占比较高的复杂环境中，随着障碍物数量的增加，优质路径的选择空间显著缩小。传统蚁群算法所规划的路径较为冗长，呈现出明显的曲折和绕行特征，未能充分利用直线路径，导致路径效率

较低。而粒子群算法虽具有全局搜索能力，但在障碍物密集场景中，路径仍表现出较多的绕行和不必要的弯曲，暴露出其在局部调整方面的不足，导致路径长度较长且规划效率不理想。相比之下，改进算法通过引入自适应信息素更新策略和人工势场法，显著提高了路径规划效果，所规划的路径不仅更短且更为平滑，成功避开了不必要的绕行，实现了更高效的路径规划。

表 4-4 算法性能参数对比

算法	路径长度/m	运行时间/s
粒子群算法	34.971	5.416
传统蚁群算法	34.223	5.143
本文算法	29.295	4.375

同样为了减少实验的偶然性，每种算法均运行 10 次并取平均值，结果如表 4-4 所示。改进算法的平均路径长度较传统蚁群算法缩短了 4.928 米，相比粒子群算法缩短了 5.676 米。在运行时间方面，改进算法比传统蚁群算法降低了约 14.9%，比粒子群算法降低了约 19.2%。这种双重优化得益于改进算法结合了自适应信息素更新策略和人工势场法，使其在复杂环境中能够快速收敛到更优解，同时有效减少不必要的计算开销。相比之下，粒子群算法在路径长度和运行时间上表现最差，原因在于其全局搜索机制在障碍物密集的场景中缺乏足够的局部精细调整能力，从而导致效率较低。总体而言，改进算法不仅路径更短、运行更快，还表现出较强的适应性和收敛能力。

4.5.4 B 样条曲线平滑

B 样条曲线是一种由控制点定义参数化的曲线，通过局部基函数加权组合控制点来形成曲线。构建时，需设定控制点和节点向量，控制点决定曲线形状，节点向量是非递减序列，限制基函数的影响范围。曲线通过基函数加权控制点形成，每个控制点仅影响局部区域，基函数确定影响范围。B 样条曲线具有优秀的局部控制特性，修改控制点仅影响局部段，整体形状不变。

$$C(u) = \sum_{i=0}^n P_i N_{i,k}(u) \quad (4-20)$$

式(4-20)是 B 样条曲线的数学表达式, 式中 P_i 是控制点, 共有 $n+1$ 个, k 为曲线的次数, $N_{i,k}(u)$ 为 k 次 B 样条基函数, u_i 为节点, 定义如式(4-21)所示:

$$\begin{cases} N_{i,1}(u) = \begin{cases} 1 & , u_i \leq u \leq u_{i+1} \\ 0 & , otherwise \end{cases} \\ N_{i,k}(u) = \frac{(u-u_i)N_{i,k-1}(u)}{u_{i+k-1}-u_i} + \frac{(u_{i+k}-u)N_{i+1,k-1}(u)}{u_{i+k}-u_{i+1}} \end{cases} \quad (4-21)$$

图 4-10 中展示了原始路径和经过三次 B 样条曲线优化后的路径对比。蓝色的直线代表的是未经优化的原始路径, 路径中包含了较多的急转弯, 可能导致无人车无法运行。通过对路径进行三次 B 样条曲线优化后, 路径变得平滑, 形成了一条红色的平滑曲线。三次 B 样条曲线优化通过引入平滑的控制点, 使得路径在整体上更具可行性。

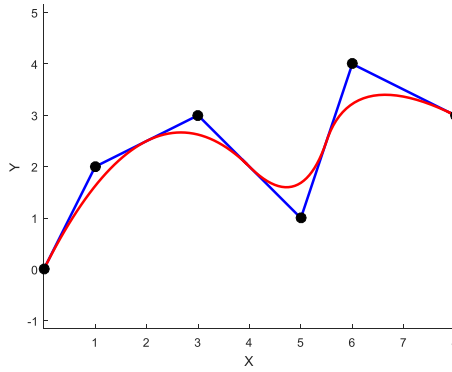


图 4-10 三次 B 样条曲线平滑路径图

4.5.5 融合交通标志的环境仿真结果

通过在不同数量障碍物环境中的对比实验, 验证了改进蚁群算法在路径规划中的优越性。然而, 实际应用中还应遵守交通规则, 因此本文将交通标志作为约束条件, 让无人车规划从起点到终点的路径, 从而对比分析改进算法在有无交通标志约束下的表现。

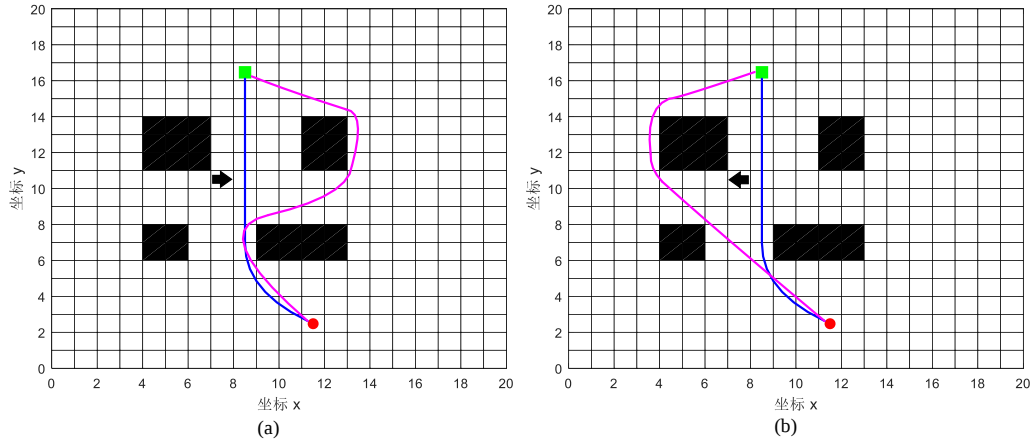


图 4-11 存在交通标志环境的仿真实验结果 (a)右转交通标志 (b)左转交通标志

在图 4-11 中，红色圆形标记表示起点，绿色方形标记表示终点，黑色区域为障碍物，箭头标记表示左右转指示。蓝色路径是改进后的蚁群算法在忽略交通标志指引时规划的最优路径，红色路径是遵循交通标志指引的最优路径。在忽略交通标志的情况下，改进算法能够在最短路径上找到从起点到终点的最佳路线，并保持与障碍物的安全距离。而在遵循交通标志指引时，改进算法不仅有效避开障碍物，还在符合交通标志的前提下规划出最短路径。虽然遵循标志的路径稍长，但仍是最佳路径，并符合实际道路交通规则。实验结果表明，改进后的蚁群算法不仅保证了路径质量，还能够高效遵循交通指示，实现无人车在复杂环境中的安全高效路径规划。

4.6 本章小结

本章针对无人车路径规划中传统算法的不足，提出了一种基于改进蚁群算法与 AAPF 混合的路径规划算法。该算法通过引入非均匀初始信息素分布、自适应信息素更新策略及惩罚系数，显著提升了算法的全局搜索能力、收敛速度和路径优化效果。同时，改进的人工势场法有效克服了目标不可及及局部极小点问题。仿真实验在不同障碍物密度的栅格地图中验证了该算法的优越性，结果表明其在路径长度、运行时间及避障性能方面均优于传统蚁群算法。此外，为满足实际应用需求，引入交通标志作为约束条件，并采用三次 B 样条曲线对路径进行平滑处理，从而进一步提升了算法在复杂交通环境中的适应性与实用性。

第 5 章 实验与分析

第三章和第四章分别介绍了基于改进 YOLOv8s 的交通标志检测算法和改进蚁群混合 AAPF 的路径规划算法。本章将这两种改进算法集成到 HW-Autocar 无人车平台，并通过实验验证其实际应用效果。实验设计了左转与右转交通标志场景，测试了系统在环境地图构建、标志检测和自主导航中的表现。实验结果表明，改进后的算法能够精准识别交通标志并规划合理路径，验证了所提方法的有效性。

5.1 无人车实验平台搭建

5.1.1 无人车硬件结构

本实验所用的智能小车是 HW-Autocar 移动机器人，如图 5-1 所示，车身长 56cm、宽 35cm、高 23cm。在硬件方面，HW-Autocar 配备了华为昇腾 310 系列 AI 处理器，集成了一个主频为 500MHz 的 DaVinciV300 AI 核心和四个主频为 1.0GHz 的 TAISHANV200M 处理器核，提供高达 8TOPS 的计算能力，极大提升了图像处理速度。无人车前端配备了一款 720P 高清摄像头，支持每秒 60 帧的视频捕捉，能够满足交通标志采集的需求。车辆还配备了 360 度全方位扫描激光雷达，扫描频率为 10Hz，测距范围为 25 米，解析度为 0.1%，能够精确感知周围环境。此外，车上还配备了高精度 IMU（惯性测量单元），其姿态角动态精度为 0.5 度，航向角动态精度为 2 度，分辨率为 0.1 度，能够为路径规划和姿态控制提供准确的实时数据。

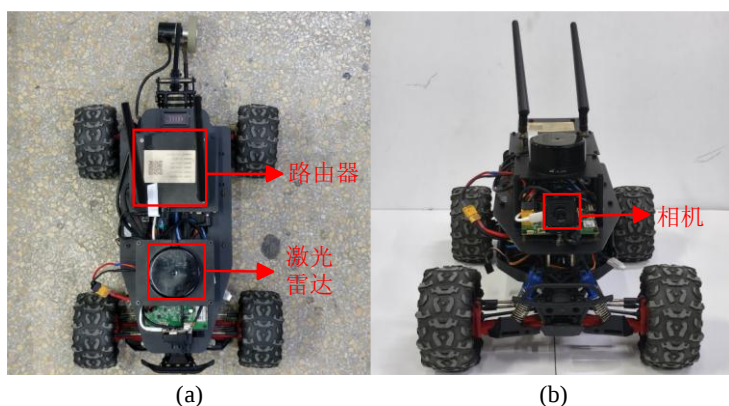


图 5-1 HW-Autocar 智能车实物图 (a)无人车俯视图 (b)无人车正面图

5.1.2 无人车软件环境

在软件环境方面，HW-Autocar 运行在 Ubuntu 22.04 操作系统上，搭载 ROS2_humble 作为机器人操作系统，提供强大的硬件控制和算法支持。ROS2_humble 是 Robot Operating System(ROS)系统的最新版本，专为高性能和实时应用设计，具有更高的可靠性和安全性，能够充分发挥 HW-Autocar 的计算能力和感知能力。该系统支持多种编程语言，包括 Python 3.8、C++和 Java，提供灵活的接口和开发工具，方便开发人员快速实现各种功能和算法。

HW-Autocar 的软件架构主要由两个模块组成：建图导航系统和通信系统。建图导航系统基于 ROS_navigation 和 ROS_mapping 两个核心模块，并集成了改进的路径规划算法，能够实现环境地图构建、全局路径规划和局部路径优化。该系统通过实时传感器数据处理，动态更新地图，保证了路径规划的精确性和实时性。

5.1.3 主从机通信机制

HW-Autocar 的主从机通信机制依赖于网络连接和 SSH 协议。PC 主机与无人车之间的通信主要通过无线网络实现，无人车配备了 Wi-Fi 路由器，主机通过 Wi-Fi 连接到无人车，并确保它们处于同一局域网内。通信的实现基于 SSH 协议，确保主机与无人车之间的安全、稳定的数据传输。

当主机与无人车在同一局域网下时，主机通过终端执行命令“ssh-racecar@智能车的 ip 地址”来建立与无人车的远程通信连接。该命令通过 SSH 协议使主机能够与无人车的操作系统进行交互，实现了远程控制和数据交换。通信结构如图 5-2 所示

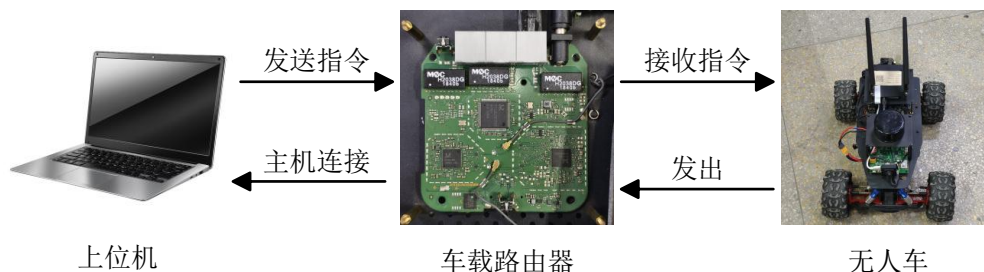


图 5-2 HW-Autocar 智能车通信结构

5.2 实验设计流程

实验在室内搭建了包含交通标志的场景，流程分为三个阶段：地图生成、检测与规划，以及无人车自主运行。首先，通过激光雷达采集环境信息，构建环境地图并标定起点与终点。接着，通过摄像头获取视频数据，并利用改进的 YOLOv8s 算法检测交通标志，结合地图信息进一步完善数据。然后，执行路径规划算法，基于障碍物和环境信息计算从起点到终点的最优路径。完成路径规划后，编译模型并通过 Wi-Fi 将可执行代码传输至无人车，最后小车根据指令执行运动，完成实验。实验流程如图 5-3 所示。

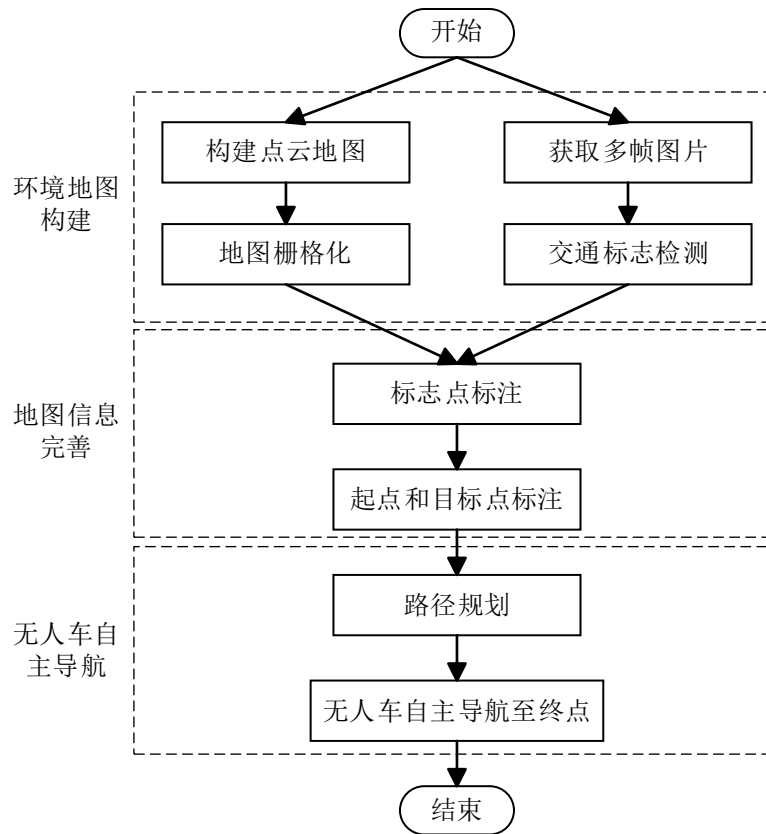


图 5-3 无人车实验流程图

5.2.1 环境地图构建

在环境地图构建过程中，主机 PC 首先通过 Wi-Fi 与无人车建立连接，随后将无人车置于实验环境的起始位置（此位置同时作为后续路径规划的起点）。通过主机发送终端命令，控制无人车在实验环境中移动，尽可能覆盖整个区域，以

获取更全面的环境数据。在上位机上，通过 ROS 中的 RVIZ 工具可以实时监控环境地图的构建过程，确保所有物品均被准确呈现。地图构建完成后，输入保存指令，即可将地图以 pgm 格式存储。图 5-4 展示了实际环境及其构建的地图。

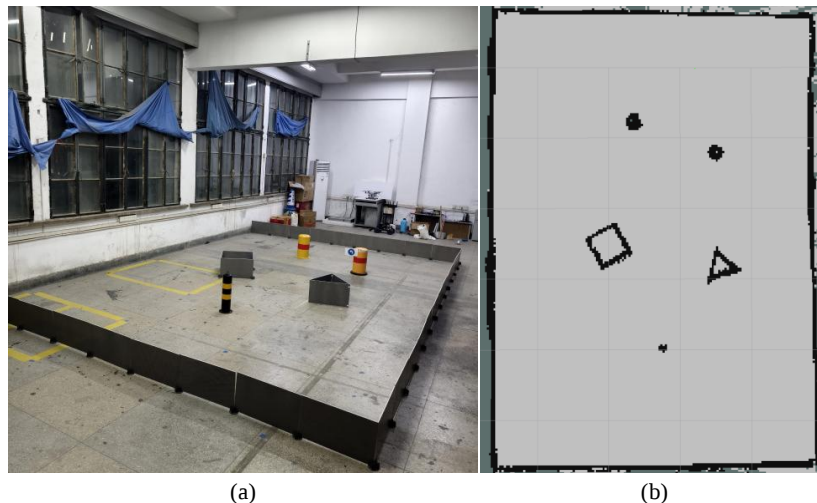


图 5-4 (a)实际环境 (b)2D 地图

5.2.2 地图信息完善

在构建环境地图过程中，激光雷达用于扫描环境中的物体并确定其位置，而相机则负责识别物体类别。通过两者数据融合，能够精确定位各物体的空间位置。首先，将环境地图进行栅格化处理；随后，利用激光雷达与相机间的坐标转换参数，计算出交通标志的空间坐标。这些数据被作为路径规划算法的输入，其中交通标志的位置对规划结果具有约束作用。如图 5-5 所示，展示了完整的地图信息。



图 5-5 完善信息地图

5.2.3 无人车自主导航

在无人车自主导航中，路径规划至关重要。为提升其精度与实时性，本文提出了一种改进的蚁群路径规划方法。路径规划过程分为两部分：首先，当检测到交通标志时，在其一定距离位置设定一个阶段目标点。第一部分是从起点到阶段目标点的路径规划，第二部分是从阶段目标点出发，按照交通标志指引的方向规划至终点的路径。这两部分路径结合，构成了从起点到终点的完整路径。

路径规划算法首先读取工作空间中的地图数据，并根据设定的安全距离和障碍物信息进行规划。规划结果以坐标点形式输出，并存储在数组中。接着，路径规划模块采用改进后的蚁群算法在 ROS 系统中运行，确保路径规划不仅避开障碍物，还能考虑安全距离。最后，启动导航功能，将无人车放置在起点，按规划路径行驶至终点，完成实验。

5.3 实验结果分析

本实验共设置两个测试场景，以验证改进算法在不同交通标志引导下的路径规划效果。两个场景的障碍物布局保持不变，但交通标志的种类不同。场景一采用左转交通标志，场景二使用右转交通标志，并额外设置无交通标志的对照实验。场景一：

在场景一的环境信息采集过程中，无人车搭载的摄像头采集视频数据，并从中截取展示车辆运行状态的关键画面，如图 5-6 所示。

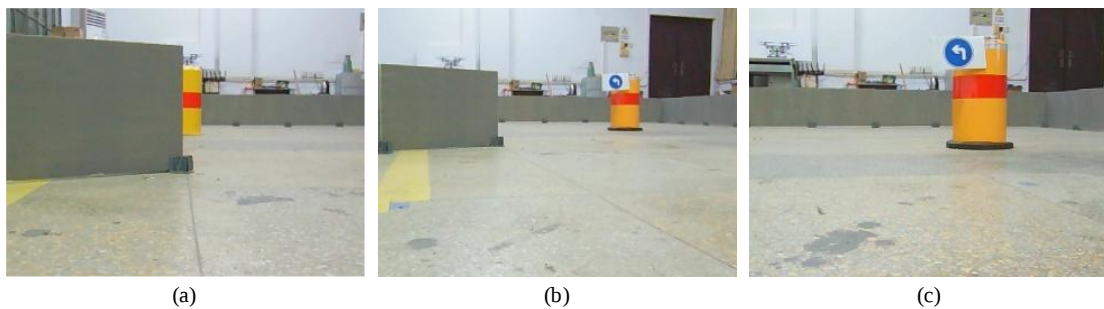


图 5-6 无人车采集的部分图片 (a)无交通标志 (b)远距离交通标志 (c)近距离交通标志

随后，基于部署的改进 YOLOv8s 算法对交通标志进行检测，并在路径规划过程中识别交通标志，获取检测结果，如图 5-7 所示，对于包含交通标志的图像，YOLOv8s 模型能够精准定位并用矩形框标注交通标志的位置，对于不含交通标

志的图像，模型不会产生误检或错误标注，而是直接返回空结果，表明图像中未检测到目标。

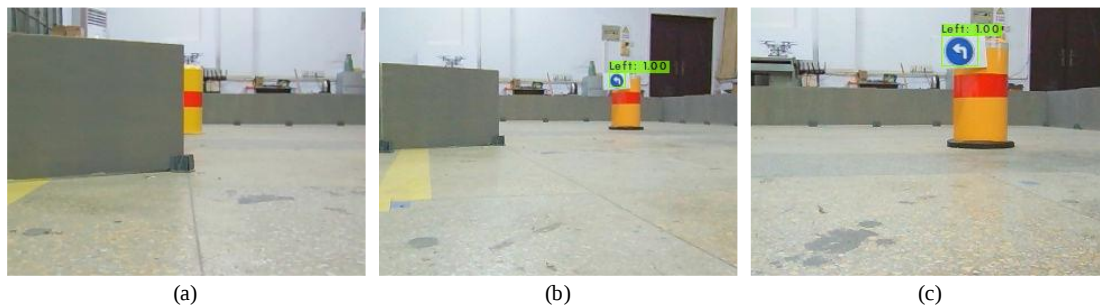


图 5-7 交通标志检测结果 (a)无交通标志 (b)、(c)左转交通标志

如图 5-8 所示，展示了无人车在路径规划过程中的部分关键帧。从图中可以观察到，无人车在避障行驶过程中能够根据交通标志的指引做出合理的路径调整。在(a)-(c)图中，无人车检测到前方的左转标志后，逐步调整自身姿态，并向左转弯以完成路径规划调整。在(d)-(f)图中，无人车成功避开障碍物，并按照规划路径顺利前进。这表明本文提出的改进蚁群混合 AAPF 路径规划算法能够结合交通标志信息，实现更加智能和安全的路径规划。

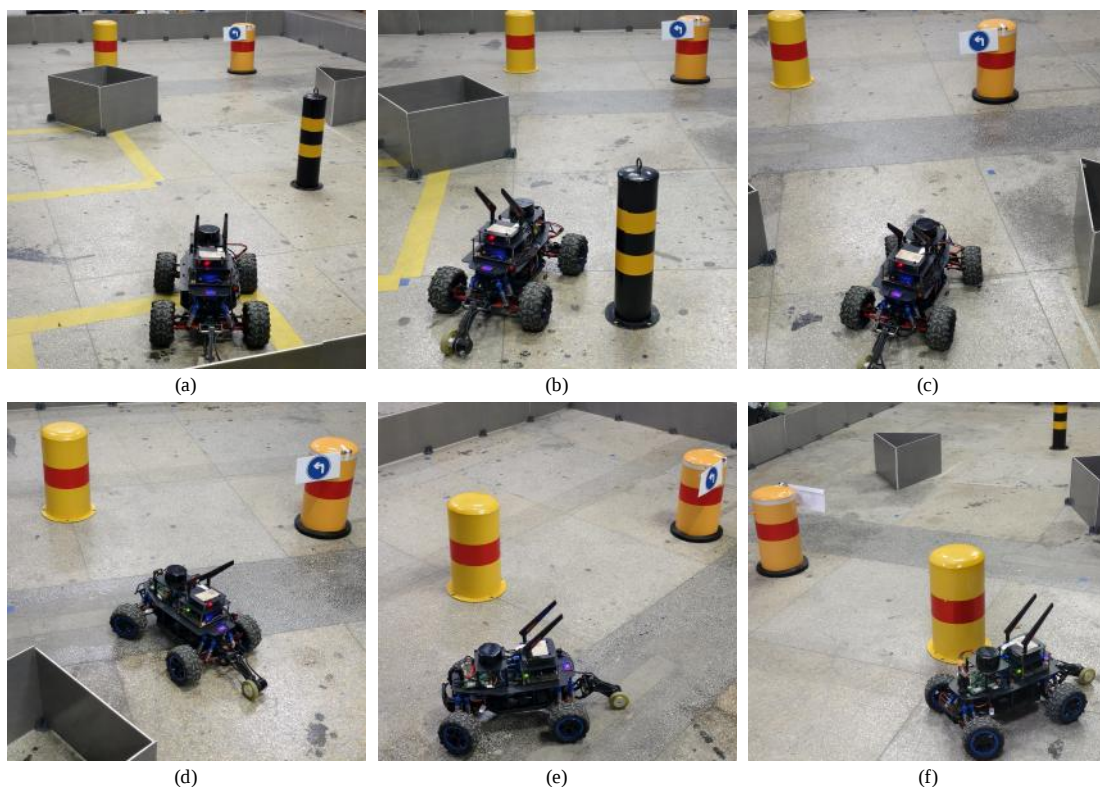


图 5-8 无人车路径规划过程部分截图

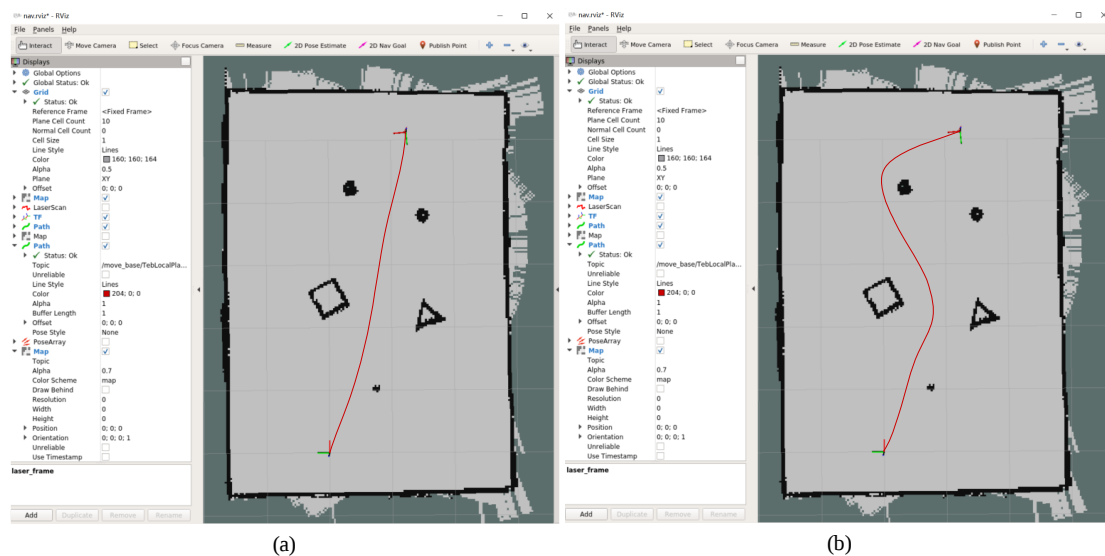


图 5-9 无人车路径规划结果 (a)无交通标志 (b)左转交通标志

从图 5-9 可以看出,在无交通标志的情况下,无人车根据改进的路径规划算法生成的路径较为直接,沿着最优路径避障前进。而在有交通标志的情况下,无人车在行驶过程中检测到左转标志,会在交通标志附近设定局部目标点,并采用改进的蚁群混合 **AAPF** 路径规划算法,先规划起点至局部目标点的路径,再规划局部目标点至终点的路径,最终在交通标志处实现左转,使无人车在遵循交通规则的同时,依然能够顺利抵达目标点,确保路径规划的合理性。

场景二：

在场景二的环境信息采集过程中,也利用无人车搭载的摄像头获取视频,并从中截取部分 RGB 图像,如图 5-10 所示。

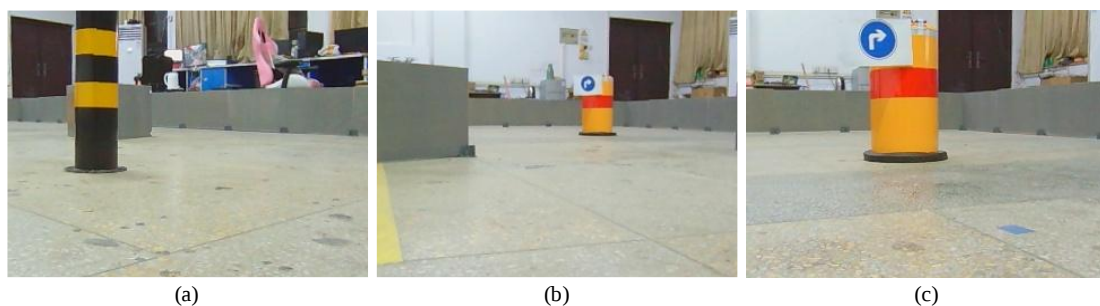


图 5-10 无人车采集的部分图片 (a)无交通标志 (b)远距离交通标志 (c)近距离交通标志

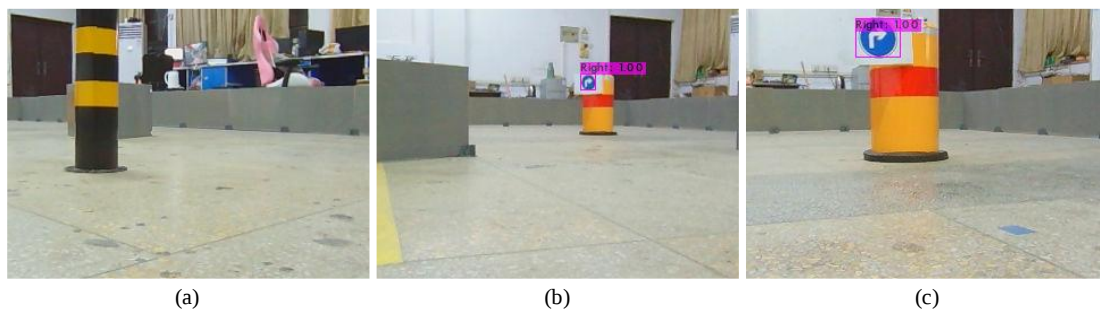


图 5-11 交通标志检测结果 (a)无交通标志 (b)、(c)右转交通标志

如图 5-12 所示，展示了无人车在路径规划过程中的部分截图。在路径规划过程中，无人车在较远处即可识别图 5-9(b)中绿色正方形位置的交通标志，并判断其类别为右转标志。依据检测结果，算法在交通标志前方适当位置设置局部目标点，随后采用改进的蚁群混合 AAPF 路径规划算法，分步骤进行路径规划：首先规划从起点至局部目标点的路径，再规划局部目标点到终点的路径，最终合并形成符合交通规则且平稳可行的路线。在此过程中，交通标志的检测结果如图 5-11 所示，YOLOv8s 模型展现出了优秀的检测能力，能够精准识别交通标志的位置，并利用矩形框进行标注。

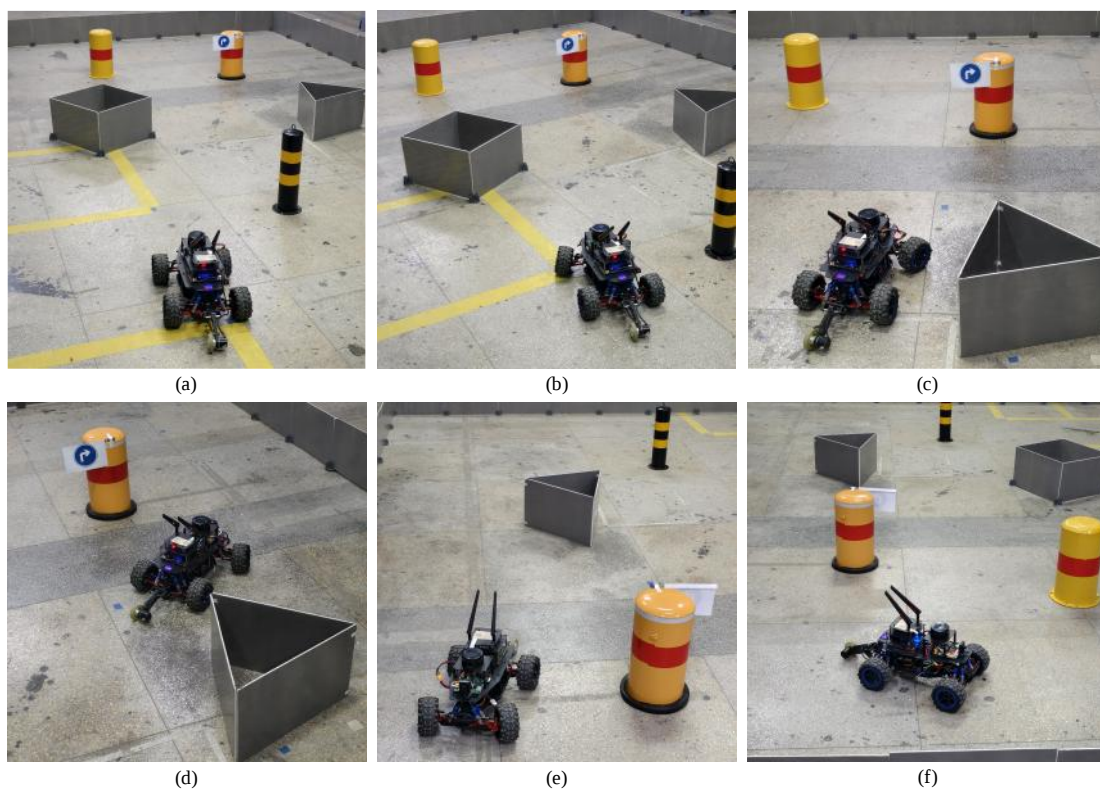


图 5-12 无人车路径规划过程部分截图

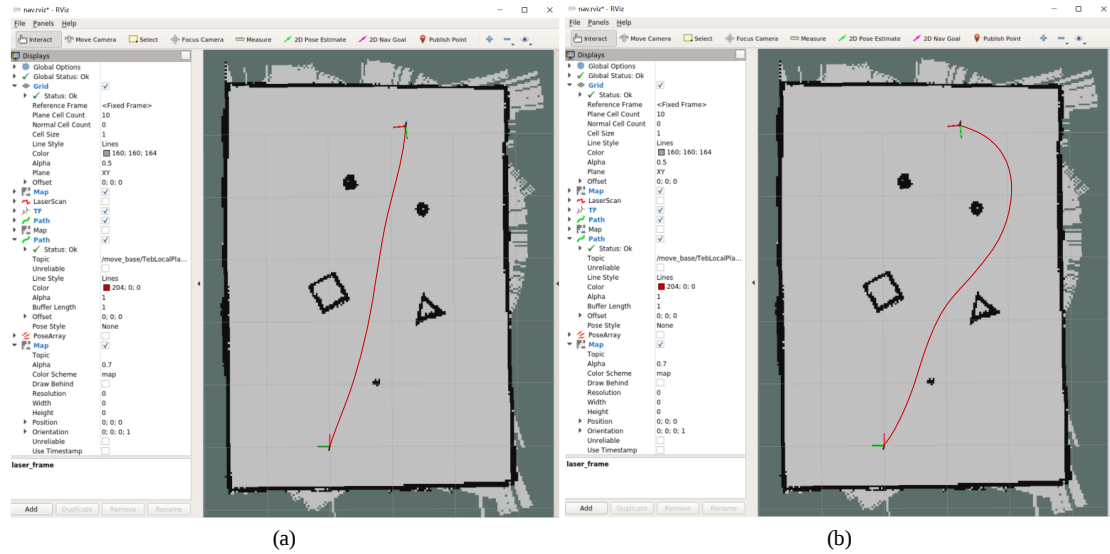


图 5-13 无人车路径规划结果 (a)无交通标志 (b)右转交通标志

从图 5-13 可以看出，无人车在路径规划过程中因交通标志的存在与否展现出不同的行驶策略。在无交通标志的情况下，无人车利用改进的路径规划算法生成了一条较为直接的路径，路径以直线为主，成功避开了沿途的障碍物，体现了算法在无外部规则约束时优先选择最短路径的高效性。而在有交通标志的环境下，无人车识别到右转标志后，算法在标志附近设定局部目标点，并采用分阶段规划策略，形成在交通标志位置完成右转的合理路径，同时确保绕避障碍物。

通过上述两个场景的测试，改进后的 YOLOv8s 交通标志检测算法无论目标远近，都能精准识别交通标志；同时，改进的蚁群混合 AAPF 路径规划算法在交通标志约束下，能够迅速重新规划出最优路径。本文成功将这些改进算法集成部署于无人车平台，并圆满完成测试，充分证明了改进算法在实际场景中的高效性与可靠性。

5.4 本章小结

本章将改进的 YOLOv8s 交通标志检测算法与改进的蚁群混合 AAPF 路径规划算法集成到 HW-Autocar 无人车上，并在不同场景中进行了实验验证。结果表明，改进的 YOLOv8s 算法能够精准识别交通标志，改进的蚁群混合 AAPF 算法则根据检测结果灵活规划合理路径，实现避障与规则遵循的平衡。实验验证了所提方法在实际场景中的有效性与可行性。

第 6 章 总结与展望

6.1 主要工作总结

在无人车领域，交通标志检测和路径规划一直是研究中的关键难点。为解决交通标志漏检和误检问题，本文提出了一种改进的 YOLOv8s 交通标志检测算法；同时，在交通标志约束下，为实现快速规划出既平滑又具备强大避障能力的路径，本文设计了一种改进的蚁群混合 AAPF 路径规划算法。基于这些方法，本文在真实环境中利用无人小车进行了实验验证。本文主要工作涵盖以下三个方面：

(1) 交通标志检测部分

针对交通标志检测问题，提出了一种基于 YOLOv8s 的改进模型。该模型融合轻量级 RepGhost 和 Bottleneck 模块，并用 RepGhost Bottleneck 模块替换所有 Bottleneck 结构，从而降低网络复杂度。为提升小目标检测能力，新增专门针对小目标的检测头，增强模型在复杂背景下识别小型交通标志的性能。同时，在主干网络引入 CBAM 注意力机制，提高对目标区域的关注度。此外，采用 MPDIoU 损失函数替代原有 CIoU 损失函数，加速网络收敛。实验在 TT100K 数据集上验证了该模型，在保持高准确率和召回率的同时，能有效检测图像中的小型交通标志。

(2) 路径规划部分

针对路径规划算法存在的局部最优、目标不可达及局部极小点等问题，本文提出了一种改进的蚁群混合 AAPF 路径规划算法。该算法通过自适应调整信息素更新策略，提高全局搜索能力并加快收敛速度，同时引入惩罚机制，增强跳出局部最优的能力。此外，与 AAPF 算法融合，有效解决了目标点位于障碍物附近时路径不可达的问题。在全局路径规划完成后，本文采用三次 B 样条曲线对路径进行平滑优化，使路径更加流畅，符合无人车的运动学约束和行驶习惯，减少轨迹突变导致的控制难度。同时，将交通标志检测结果融入路径规划，使无人车能够根据交通标志信息实时调整路径，提高规划的合理性和安全性，尤其在变道、避障和交叉路口等复杂场景下具备更强的适应能力。仿真结果表明，本文提出的

算法在路径可行性、平滑性和避障能力方面均优于传统算法，验证了其有效性和优越性。

(3) 实验验证部分

为验证改进算法在实际无人车系统中的有效性与可行性，本文将改进的 YOLOv8s 交通标志检测算法与改进的蚁群混合 AAPF 路径规划算法集成部署至 HW-Autocar 无人车平台。在不同场景下进行实验，实验结果表明，改进算法能够准确检测交通标志，并根据检测结果灵活调整行驶路径，同时实现快速避障和稳定的全局路径规划。

6.2 进一步工作展望

本文在基于交通标志检测的无人车路径规划领域取得了一定进展，但是无人车技术涵盖多个领域，涉及面广，工作量较大。受实验室环境和设备限制，本研究仍有诸多需要改进和完善之处。

(1) 本文的目标检测算法主要应用于常规场景，尚未涉及室外环境及恶劣条件下的鲁棒性问题。后续研究需进一步拓展至更复杂的应用场景，重点提升算法在多变环境下的适应能力。例如，在室外光照变化、恶劣天气影响以及复杂背景干扰等情况下，目标检测的稳定性和精度仍有待优化。

(2) 针对实际场景中的动态障碍物，未来可引入预测模型，通过对障碍物运动轨迹的分析，提前规划避让策略，以提高无人车在动态环境下的路径规划效率。

(3) 目前研究主要针对单辆无人车在交通标志约束下的路径规划，而在实际应用中，无人车通常需要协同作业以应对复杂交通场景，例如车队运输、拥堵路段的协作避让或多车任务分配。未来需扩展至多智能车系统的协作路径规划，研究多车间的通信与协调机制，以确保在多车环境下路径规划算法能够实现高效避碰和信息互通。

参考文献

- [1] 张道文, 母尧尧, 王朝健, 等. 城市道路交通事故特性及严重程度研究[J]. 安全与环境学报, 2022, 22(02): 599-605.
- [2] Aggarwal S, Kumar N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges[J]. Computer communications, 2020, 149: 270-299.
- [3] Wang C, Mao J. Summary of agv path planning[C]//2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE). IEEE, 2019: 332-335.
- [4] Mielle M, Magnusson M, Lilienthal A J. A comparative analysis of radar and lidar sensing for localization and mapping[C]//2019 European Conference on Mobile Robots (ECMR). IEEE, 2019: 1-6.
- [5] Iftikhar S, Asim M, Zhang Z, et al. Target detection and recognition for traffic congestion in smart cities using deep learning-enabled UAVs: A review and analysis[J]. Applied Sciences, 2023, 13(6): 39951-39976.
- [6] 王科俊, 赵彦东, 邢向磊. 深度学习在无人驾驶汽车领域应用的研究进展[J]. 智能系统学报, 2018, 13(1): 55-69.
- [7] Wang Z, Wang M. Development status and challenges of unmanned vehicle driving technology[J]. Big Data Analytics for Cyber-Physical System in Smart City: BDCPS 2019, 28-29 December 2019, Shenyang, China, 2020: 68-74.
- [8] Kasap A. The historical development of AVs[M]//Autonomous Vehicles. Edward Elgar Publishing, 2022: 7-14.
- [9] Chen G, Zhang W. Design of prototype simulation system for driving performance of electromagnetic unmanned robot applied to automotive test[J]. Industrial robot: an international journal, 2015, 42(1): 74-82.
- [10] 贾锦云. 面向无人驾驶的车道保持智能决策控制方法研究[D]. 北京, 北京交通大学, 2021.

-
- [11] 李德毅, 高洪波. 基于驾驶脑的智能驾驶车辆硬件平台架构[J]. Engineering, 2018, 4(04): 78-92.
 - [12] 王浩宇, 左志强, 邓宇翔, 等. 无人驾驶车辆运动控制综述[J]. 控制理论与应用, 2024, 41(09): 1507-1522.
 - [13] 徐进, 陈钦, 陈正委, 等. 适应无人驾驶汽车的道路设施设计综述[J]. 西南交通大学学报, 2023, 58(06): 1366-1377.
 - [14] Liu C, Li S, Chang F, et al. Machine vision based traffic sign detection methods: review, analyses and perspectives[J]. IEEE Access, 2019, 7: 86578-86596.
 - [15] Li H, Sun F, Liu L, et al. A novel traffic sign detection method via color segmentation and robust shape matching[J]. Neurocomputing, 2015, 169: 77-88.
 - [16] Xu X, Jin J, Zhang S, et al. Smart data driven traffic sign detection method based on adaptive color threshold and shape symmetry[J]. Future generation computer systems, 2019, 94: 381-391.
 - [17] Hechri A, Hmida R, Mtibaa A. Robust road lanes and traffic signs recognition for driver assistance system[J]. International journal of computational science and engineering, 2015, 10(1-2): 202-209.
 - [18] Yan Y. Using the improved SSD algorithm to motion target detection and tracking[J]. Computational Intelligence and Neuroscience, 2022, 2022(1): 1-10.
 - [19] Jiang P, Ergu D, Liu F, et al. A review of YOLO algorithm developments[J]. Procedia computer science, 2022, 199: 1066-1073.
 - [20] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]//roceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
 - [21] Chu J, Zhang C, Yan M, et al. TRD-YOLO: a real-time, high-performance small traffic sign detection algorithm[J]. Sensors, 2023, 23(8): 3871-3884.
 - [22] Dewi C, Chen R C, Jiang X, et al. Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V4[J]. Multimedia Tools and Applications, 2022, 81(26): 37821-37845.
 - [23] 郝博, 尹兴超, 闫俊伟, 等. 基于 Gan-St-YOLOv5 的复杂环境下的手势识别

- [J]. 东北大学学报(自然科学版), 2023, 44(07): 953-963.
- [24] Chen Z, Liu C, Filaretov V F, et al. Multi-scale ship detection algorithm based on YOLOv7 for complex scene SAR images[J]. Remote Sensing, 2023, 15(8): 2071-2088.
- [25] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
- [26] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
- [27] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(6): 1137-1149.
- [28] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//ceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
- [29] Cao J, Zhang J, Jin X. A traffic-sign detection algorithm based on improved sparse r-cnn[J]. IEEE Access, 2021, 9: 122774-122788.
- [30] Zhang J, Xie Z, Sun J, et al. A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection[J]. IEEE access, 2020, 8: 29742-29754.
- [31] Qiao L, Zhao Y, Li Z, et al. Defrcn: decoupled faster r-cnn for few-shot object detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 8681-8690.
- [32] 马啸, 邵利民, 金鑫, 等. 基于改进 Mask R-CNN 的可见光图像中舰船目标检测方法[J]. 北京理工大学学报, 2021, 41(07): 734-744.
- [33] 余修武, 穆静, 刘永. 基于遗传模拟退火优化的 DV-Hop 定位算法[J]. 华中科技大学学报(自然科学版), 2024, 52(03): 149-155.
- [34] Luo J, Wang Z X, Pan K L. Reliable path planning algorithm based on improved artificial potential field method[J]. IEEE Access, 2022, 10: 108276-108284.
- [35] Garud K S, Jayaraj S, Lee M Y. A review on modeling of solar photovoltaic systems

- p>using artificial neural networks, fuzzy logic, genetic algorithm and hybrid models[J]. International Journal of Energy Research, 2021, 45(1): 6-35.
- [36] Balan K, Luo C. Optimal trajectory planning for multiple waypoint path planning using tabu search[C]//2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, 2018: 497-501.
- [37] Shi K, Wu Z, Jiang B, et al. Dynamic path planning of mobile robot based on improved simulated annealing algorithm[J]. Journal of the Franklin Institute, 2023, 360(6): 4378-4398.
- [38] Pandey A, Sonkar R K, Pandey K K, et al. Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller[C]//2014 IEEE 8th international conference on intelligent systems and control (ISCO). IEEE, 2014: 39-41.
- [39] Deng W, Xu J, Zhao H. An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem[J]. IEEE access, 2019, 7: 20281-20292.
- [40] Wang G, Zhou J. Dynamic robot path planning system using neural network[J]. Journal of Intelligent & Fuzzy Systems, 2021, 40(2): 3055-3063.
- [41] Jia H, Wei Z, He X, et al. Path planning based on improved particle swarm optimization algorithm[J]. 2018, 49(12): 371-377.
- [42] Lamini C, Benhlima S, Elbekri A. Genetic algorithm based approach for autonomous mobile robot path planning[J]. Procedia Computer Science, 2018, 127: 180-189.
- [43] Liu X, Zhang D, Zhang J, et al. A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm[J]. Cluster Computing, 2021, 24(3): 1901-1915.
- [44] Tuncer A, Yildirim M. Dynamic path planning of mobile robots with improved genetic algorithm[J]. Computers & Electrical Engineering, 2012, 38(6): 1564-1572.
- [45] Qing G, Zheng Z, Yue X. Path-planning of automated guided vehicle based on improved Dijkstra algorithm[C]//2017 29th Chinese control and decision conference (CCDC). IEEE, 2017: 7138-7143.

-
- [46] Persson S M, Sharf I. Sampling-based A* algorithm for robot path-planning[J]. The International Journal of Robotics Research, 2014, 33(13): 1683-1708.
 - [47] Zong C, Han X, Zhang D, et al. Research on local path planning based on improved RRT algorithm[J]. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2021, 235(8): 2086-2100.
 - [48] Zhang J, Liu Z, Wang Y, et al. Research on effective path planning algorithm based on improved A* algorithm[C]//Journal of Physics: Conference Series. IOP Publishing, 2022, 1-9.
 - [49] Li C, Huang X, Ding J, et al. Global path planning based on a bidirectional alternating search A* algorithm for mobile robots[J]. Computers & Industrial Engineering, 2022, 168: 108-126.
 - [50] Li Z, Liu F, Yang W, et al. A survey of convolutional neural networks: analysis, applications, and prospects[J]. IEEE transactions on neural networks and learning systems, 2021, 33(12): 6999-7019.
 - [51] LeCun Y. Generalization and network design strategies[J]. Connectionism in perspective, 1989, 19(18): 143-155.
 - [52] Tammina S. Transfer learning using vgg-16 with deep convolutional neural network for classifying images[J]. International Journal of Scientific and Research Publications (IJSRP), 2019, 9(10): 143-150.
 - [53] Salavati P, Mohammadi H M. Obstacle detection using GoogleNet[C]//2018 8th international conference on computer and knowledge engineering (ICCCKE). IEEE, 2018: 326-332.
 - [54] Wu Z, Shen C, Van Den Hengel A. Wider or deeper: Revisiting the resnet model for visual recognition[J]. Pattern recognition, 2019, 90: 119-133.
 - [55] Apicella A, Donnarumma F, Isgrò F, et al. A survey on modern trainable activation functions[J]. Neural Networks, 2021, 138: 14-32.
 - [56] Dombi J, Jónás T. The generalized sigmoid function and its connection with logical operators[J]. International Journal of Approximate Reasoning, 2022, 143: 121-138.
 - [57] Abdelouahab K, Pelcat M, Berry F. Why tanh is a hardware friendly activation

- function for cnns[C]//Proceedings of the 11th international conference on distributed smart cameras. 2017: 199-201.
- [58] Si J, Harris S L, Yfantis E. A dynamic ReLU on neural network[C]//2018 IEEE 13th Dallas Circuits and Systems Conference (DCAS). IEEE, 2018: 1-6.
- [59] Köseoğlu M, Çelik O M, Pektaş Ö. Design of an autonomous mobile robot based on ROS[C]//2017 International Artificial Intelligence and Data Processing Symposium (IDAP). IEEE, 2017: 1-5.
- [60] Bao Z, Hossain S, Lang H, et al. A review of high-definition map creation methods for autonomous driving[J]. Engineering Applications of Artificial Intelligence, 2023, 122: 1061-1083.
- [61] Fisher A, Cannizzaro R, Cochrane M, et al. ColMap: a memory-efficient occupancy grid mapping framework[J]. Robotics and Autonomous Systems, 2021, 142: 1037-1046.
- [62] Zhang T, Zhu Y, Song J. Real - time motion planning for mobile robots by means of artificial potential field method in unknown environment[J]. Industrial Robot: An International Journal, 2010, 37(4): 384-400.
- [63] Yang X. Development of obstacle avoidance for autonomous vehicles and an optimization scheme for the artificial potential field method[C]//2021 2nd International Conference on Computing and Data Science (CDS), 2021: 12-18.
- [64] An L, Chen T, Cheng A, et al. Simulation on the path planning of intelligent vehicles based on artificial potential field algorithm[J]. Automot. Eng, 2017, 39(12): 1451-1456.
- [65] Dorigo M, Stützle T. Ant colony optimization: overview and recent advances[M]. Springer International Publishing, 2019, 311-351.
- [66] Liu J, Yang J, Liu H, et al. An improved ant colony algorithm for robot path planning[J]. Soft computing, 2017, 21: 5829-5839.
- [67] Dai X, Long S, Zhang Z, et al. Mobile robot path planning based on ant colony algorithm with A* heuristic method[J]. Frontiers in neurorobotics, 2019, 13: 9-25.

攻读硕士学位期间主要研究成果

(1) 发表论文

- [1] 王坤相, 王正刚, 王世康. 基于改进 YOLOv8 的交通标志检测算法[J]. 哈尔滨商业大学学报. (已录)
- [2] 王世康, 王坤相, 江自豪等. 基于改进 YOLOv8n 的无人机航拍图像小目标检测[J]. 传感技术学报. (已录)

(2) 获奖情况

- [1] 2023 年第十八届全国大学生智能汽车竞赛室外 ROS 无人竞速赛全国初赛一等奖
- [2] 2023 年第十八届全国大学生智能汽车竞赛室外 ROS 无人竞速赛全国总决赛一等奖(季军)
- [3] 2023 年第十八届全国大学生智能汽车竞赛创意组航天智慧物流全国初赛一等奖
- [4] 2023 年第十八届全国大学生智能汽车竞赛创意组航天智慧物流全国总决赛一等奖
- [5] 2023 年第十八届全国大学生智能汽车竞赛室外无人驾驶自行车挑战赛全国初赛二等奖
- [6] 2023 年第十八届全国大学生智能汽车竞赛室外无人驾驶自行车挑战赛全国总决赛二等奖
- [7] 2023 年第十七届“西门子杯”中国智能制造挑战赛离散行业自动化方向全国初赛二等奖
- [8] 2024 年第十九届全国大学生智能汽车竞赛创意组讯飞智慧救援全国初赛一等奖
- [9] 2024 年第十九届全国大学生智能汽车竞赛创意组讯飞智慧救援全国总决赛二等奖
- [10] 2024 年第十八届“西门子杯”中国智能制造挑战赛离散行业自动化方向全国初赛一等奖

致 谢

行文至此，落笔为终。当论文定稿的这一刻，万千思绪涌上心头。求学之路道阻且长，幸得诸多良师益友、至亲家人相伴而行，在此谨以最诚挚的笔触表达谢意。

师恩如海，衔草难报。首先要向我的导师王正刚和王冠凌老师致以最深切的感激。从选题方向的反复推敲到研究框架的搭建，从实验数据的严谨分析到终稿字句的精心打磨，您们总以渊博的学识与春风化雨般的耐心给予指导。您们不仅是学术道路的引路人，更以严谨治学的态度教会我“学问之道在求实”的真谛。

同窗情谊，幸甚至哉。感谢我的同门师兄与课题组伙伴们在课题攻坚期的并肩作战，实验室里共同调试代码的日夜，研讨会上思维碰撞的火花，我们互相鼓励的身影，都将成为青春记忆中最珍贵的片段。你们的存在让这段苦乐交织的求学时光始终充满向上的力量。

椿萱之恩，山高水长。最后，要把最深沉的谢意献给我的父母。二十余载求学路，是你们用并不宽阔的肩膀为我托起追逐理想的天空。家人们默默的支持，是我在文献堆中挑灯夜战时的最大慰藉。唯愿以此刻的薄薄论文，回报你们如山海般厚重的爱。

此外，还要感谢评审专家拨冗指正，感谢安徽工程大学。纸短情长，所有未能尽述的感动都将化作前行的动力。学术之路漫漫，我辈当继续求索，不负所托。

尚德敬学 唯实惟新

