

分类号: TP242.

密 级: 公开

学校代码: 10363

学 号: 2210330161



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题 目 基于深度学习的移动机器人
建图与导航研究

论 文 作 者 兰天

校 内 导 师 王冠凌（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） 控制工程与自动化技术

论文提交日期: 2024 年 5 月 16 日

分类号: TP242.6

单位代码: 10363

密 级: 公 开

学 号: 2200220163

题 目 基于深度学习的移动机器人建图与导航研究

英文并列

题 目 Research on Mapping and Navigation of Mobile
Robot Based on Deep Learning

学生姓名: 兰天

校内导师: 王冠凌

校外导师: 毛德平

专 业: 电子信息

研究方向: 控制工程与自动化技术

论文答辩日期: 2024.5.16

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：兰天

日期：2024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。

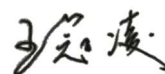
本学位论文属于

不保密 ☒。

学位论文作者签名：



指导教师签名：



日期： 2024 年 5 月 16 日

日期： 2024 年 5 月 16 日

基于深度学习的移动机器人建图与导航研究

摘要

近年来,随着移动机器人技术的不断发展,移动机器人导航的研究受到人们的重视,而在移动机器人的导航研究中,路径规划是其关键环节,一个好的路径规划算法是保证移动机器人安全高效完成任务的关键。本文采用改进的随机路径图(Probabilistic Roadmap Method, PRM)算法进行全局路径规划,同时使用深度学习与时间弹性带算法(Time Eletic Band, TEB)结合的方法进行局部路径规划。该方法旨在提升移动机器人在狭窄空间中的导航能力和搜索效率,并通过构建深度学习模型来识别和避开障碍物。本文主要研究工作如下:

(1) 针对激光 SLAM 自定位易发生错误的问题,利用扩展卡尔曼滤波算法结合姿态传感器和里程计数据来提升定位的准确性,增加了全局地图构建的成功率。此外,通过分析结合后的数据,提取出用于避障的地图、移动机器人位姿和速度等关键信息。

(2) 在移动机器人计算能力有限的情况下,为确保障碍物识别的实时性,对传统的 YOLOv5 深度神经网络模型进行改进。针对障碍物变化尺度的问题,在骨干网络的 C3 中加入 CA (Coordinate Attention) 注意力机制,增强算法对于目标的精确定位能力,针对传统 NMS 在识别重叠区域时出现遗漏和错误的问题,引入

DIoU_NMS 来保证重叠遮掩部分的精度，加入了 $g^n\text{Conv}$ 执行高阶空间交互，在原来三输出头基础上增加一个小目标检测头，提高小目标的识别精度。改进后的算法在公共数据集上测试，对比原始算法精度明显提升，并且数据量很小，适合运用到实践中。在移动机器人平台上将改进后的 YOLOv5 算法进行部署，验证改进算法的有效性。

(3) 针对 PRM 算法在狭窄空间中的局限性，对原始算法的采样策略进行改进，通过在学习阶段引入人工势场法 (Artificial Potential Field, APF)，增加狭窄区域的路径网密度，从而提高移动机器人全局路径规划的平滑性。利用 YOLOv5 算法对障碍物进行识别，根据提取出的障碍物位置信息并结合 TEB 算法及传感器数据，实现高效的实时局部避障功能。最后，融合全局和局部路径规划，并对其进行仿真与实物实验，验证本文路径规划的有效性。

关键词：深度学习，SLAM，路径规划，ROS，移动机器人

Research on Mapping and Navigation of Mobile Robot Based on Deep Learning

ABSTRACT

In recent years, with the continuous development of robot technology, the research of mobile robot navigation has been paid more attention. In the navigation research of mobile robot, path planning is the key link, and a good path planning algorithm is the key to ensure the safe and efficient completion of the task of mobile robot. In this paper, the improved Probabilistic Roadmap Method (PRM) algorithm is used for global path planning, and the combined method of deep learning and Time Eletic Band (TEB) is used for local path planning. The method aims to improve the robot's navigation ability and search efficiency in narrow Spaces, and to identify and avoid obstacles by building deep learning models. The main research work of this paper is as follows:

(1) Aiming at the problem that laser SLAM self-localization is prone to errors, extended Kalman filtering algorithm combined with attitude sensor and odometer data is used to improve the accuracy of positioning and increase the success rate of global map construction. In addition, by analyzing the combined data, key information such as map, robot pose and speed for obstacle avoidance is extracted.

(2) The traditional YOLOv5 deep neural network model is improved to ensure the real-time obstacle recognition in the case of limited computing power of mobile robots. To solve the problem of obstacle scale change, Coordinate Attention(CA) is added to C3 of the backbone network to enhance the algorithm's ability to accurately locate the target. In view of the omission and error of traditional NMS in identifying overlapping areas, DIOU_NMS is introduced to ensure the accuracy of overlapping covered parts. g^n Conv was added to carry out high-order spatial interaction, and a small target detection head was added to the original three output heads to improve the recognition accuracy of small goals. Compared with the original algorithm, the improved algorithm is tested on the public data set, and the data amount is small, which is suitable for application in practice. The improved YOLOv5 algorithm was deployed on the mobile robot platform to verify the effectiveness of the improved algorithm.

(3) In view of the limitations of PRM algorithm in narrow space, the sampling strategy of the original algorithm is improved. Artificial Potential Field (APF) is introduced in the learning stage to increase the path network density in the narrow area, so as to improve the smoothness of global path planning of mobile robots. The YOLOv5 algorithm is used to identify obstacles, and according to the extracted obstacle position information, combined with TEB algorithm and sensor data, efficient real-time local obstacle avoidance function is realized. Finally, global and local path

planning are integrated, and simulation and real experiments are carried out to verify the effectiveness of the proposed path planning.

Keywords: Deep Learning; SLAM; Path Planning; ROS; Mobile Robot

目 录

摘 要.....	III
ABSTRACT.....	V
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 移动机器人路径规划研究现状.....	2
1.2.2 深度学习在移动机器人中的发展及研究现状.....	6
1.3 论文主要研究内容与章节安排.....	8
第 2 章 移动机器人系统设计及地图构建.....	10
2.1 ROS 操作系统概述.....	10
2.1.1 ROS 发展过程.....	10
2.1.2 ROS 系统特点.....	11
2.2 移动机器人系统模型.....	12
2.2.1 坐标系系统模型.....	12
2.2.2 阿克曼转向运动学模型.....	13
2.3 激光 SLAM 地图构建.....	15
2.3.1 传统激光 SLAM.....	15
2.3.2 传统激光 SLAM 缺陷.....	17
2.4 EKF 融合里程计与 IMU 数据建图.....	18
2.4.1 IMU 原理.....	18
2.4.2 EKF 数据融合.....	19
2.5 本章小结.....	22
第 3 章 基于深度学习的目标检测.....	23
3.1 卷积神经网络概述.....	23
3.1.1 基本网络结构.....	23
3.1.2 Darknet 深度学习框架.....	27
3.2 深度学习网络结构设计.....	28
3.2.1 YOLOv5 网络模型.....	28

3.2.2 YOLOv5 模型的改进及优化.....	30
3.3 模型实验验证及结果分析.....	35
3.3.1 实验参数及准备.....	35
3.3.2 实验结果验证与分析.....	37
3.4 本章小结.....	39
第 4 章 移动机器人导航算法设计	40
4.1 基于改进 PRM 的全局路径规划算法设计	40
4.1.1 PRM 算法基本原理	40
4.1.2 PRM 算法的改进	41
4.1.3 改进的 PRM 算法测试结果与分析	42
4.2 局部路径规划算法设计.....	43
4.2.1 TEB 算法基本原理	43
4.2.2 TEB 路径规划算法仿真	45
4.2.3 TEB 算法结合深度学习避障设计	46
4.3 融合全局与局部路径规划算法.....	47
4.3.1 混合路径规划.....	47
4.3.2 混合路径规划仿真实验.....	49
4.4 本章小结.....	49
第 5 章 实验与分析.....	50
5.1 实验平台整体设计.....	50
5.1.1 硬件平台设计.....	50
5.1.2 软件功能模块设计.....	52
5.2 移动机器人深度学习模型部署.....	53
5.2.1 模型部署.....	53
5.2.2 YOLOv5 实时测试.....	53
5.3 路径规划实验.....	54
5.4 自主避障实验.....	55
5.5 本章小结.....	61
第 6 章 总结与展望.....	62

6.1 论文的主要工作.....	62
6.2 下一步研究计划.....	63
参考文献.....	64
攻读硕士学位期间主要研究成果.....	69
致 谢.....	70

第 1 章 绪论

1.1 课题研究背景及意义

自主移动机器人代表了机器人技术的高级发展阶段，它们能够在人类难以或不宜进入的恶劣环境下操作，例如在具有核辐射的环境或真空状态下进行检测和组装任务等^[1]。这种移动机器人的应用极大地减少了人们在危险环境中工作需求，从而解放了人类劳动力，在众多领域中产生了巨大的需求。自移动类机器人诞生以来，其导航能力就成为研究人员关注的焦点，导航过程涉及到多个步骤，包括环境感知、定位、路径规划以及移动执行。环境感知通常依靠摄像头、激光雷达等传感器捕获周围环境的信息，定位是移动机器人通过算法识别自己在环境中的准确位置，路径规划是基于当前位置和目标位置计算出一条可行且高效的路径，最后移动执行是机器人根据规划的路径进行实际移动，克服路径上可能存在的障碍物到达目标点。

在移动机器人的导航研究中，路径规划是其关键环节^[2]，规划从起点到终点的最优路径，同时考虑距离、时间等关键要素，以确保找到的路径成本最低，并与障碍物保持一定安全距离^[3]。随着移动机器人应用范围的不断扩大，面临的障碍也越来越多样、越来越复杂，为保证移动机器人在复杂环境下的作业效率，多种路径规划方法被相继开发和完善。当移动机器人的路径规划得当，它可以安全且迅速地到达目的地，同时缩短运行时间，节约成本和资源^[4]。路径规划方式一般分为全局和局部路径规划^[5]，全局路径规划需要掌握所有的环境信息，如初始位置、目标位置和障碍物位置，常用的算法包括基于启发式的搜索算法和各种智能算法。而局部路径规划则适用于移动机器人处于未知或不熟悉的环境中，这种情况下，移动机器人依赖于自身的传感器获取周围信息，并实时调整其移动策略。移动机器人没有预先的环境信息，仅仅依靠传感器来感知周遭环境，并基于这些实时数据从起点导航至终点。

全局路径规划和局部路径规划算法是相辅相成的技术，而不是截然不同的方法。局部路径规划在动态和复杂环境中尤为关键，这是因为它能够应对环境

的即时变化。文献^[6]结合了 A*算法与人工势场法，试验表明混合规划策略的效果更加卓越。随着深度学习理论的飞速发展，作为其核心应用之一的目标识别技术已经受到了大量关注。移动机器人技术与深度学习的结合，推动了两者的共同发展，预示着这一领域将会有巨大的发展潜力。

1.2 国内外研究现状

1.2.1 移动机器人路径规划研究现状

路径规划一般分为全局路径规划和局部路径规划，且路径规划的核心基于算法。

(1) 全局路径规划

全局路径规划是移动机器人导航系统中的一个关键组成部分，其目的是在已知环境中生成一条从起点到终点的最优路径。它考虑了所有可能的障碍物位置和环境布局，使用算法寻找成本最低的路径。这种规划通常在移动机器人任务开始前进行，以确保预定路径的有效性。尽管全局路径规划能够提供详尽的路线规划，但由于计算复杂度较高，不适合动态环境或需要实时更新的场景。

1) Dijkstra 算法

Dijkstra 算法^[7]是一种经典的基于图搜索的路径规划算法，它适用于带权重的图，其中权重代表路径成本且权重必须为非负值。算法的核心思想是贪心策略：在每一步选择最小成本的顶点，并更新该顶点邻居的最短路径估计。具体过程包括维护两个集合，一个包含已经评估过的顶点，另一个包含尚未评估的顶点。通过这种方法，Dijkstra 算法能够逐渐找出到所有顶点最短的路径。该算法广泛应用于各种领域，如网络路由、地图导航等。文献^[8]和文献^[9]展示了该技术在无人驾驶汽车中的应用，证实了其在实际运行中的适应性，并在多种情况下显示出较强的鲁棒性。尽管 Dijkstra 算法在静态图中表现优异，但在动态图或需要频繁更新的情况下效率较低，因为每次环境变化都需要重新计算。

2) A*算法

A* (A-star) 算法^[10]是广泛应用的路径规划算法之一，具有简单、稳定性高的优点，它广泛应用于游戏中的 NPC（非玩家角色）路径寻找和实际

世界的移动机器人路径规划。 A^* 算法实时更新路径的优先级，优先拓展预估总代价最低的路径，直到找到目标点为止， A^* 算法因其高效率和较少的路径开销而备受推崇。算法代价函数为：

$$f(v) = g(v) + h(v) \quad (1-1)$$

其中 $g(v)$ 是从起始点到当前节点 n 的实际代价， $h(v)$ 是起始点到当前节点 n 的估计代价（启发函数），这个启发式函数是 A^* 算法的核心，它影响算法的效率和精确性。理想情况下， $h(v)$ 应该是一种对实际距离的无偏估计，如果 $h(v)$ 总是低估实际成本，算法保证能找到最短路径，这种启发式称为可采纳的。

在执行过程中， A^* 算法维持两个集合，开放集合（Open Set）和封闭集合（Closed Set），开放集合包含待评估的节点，而封闭集合包含已经评估的节点。算法从起始节点开始，将其添加到开放集合中。算法的每一步会从开放集合中选择具有最低 $f(v)$ 值的节点进行扩展，然后生成它的邻居并计算它们的 $f(v)$ 值，将这些邻居添加到开放集合中。如果一个邻居节点已经在封闭集合中，或者新的 $g(v)$ 值更高（意味着找到了一个更差的路径），那么这个邻居会被忽略，否则这个邻居将被添加到开放集合中，等待进一步的评估。一旦目标节点被添加到封闭集合，算法就找到了一条可行路径，如果开放集合为空，说明没有路径可到达目标节点。 A^* 算法的效率和精确性依赖于启发函数 $h(v)$ 的质量，一个精心设计的启发式可以大大减少搜索范围，使得算法迅速找到最佳路径。

由于其平衡了效率和准确性， A^* 算法在路径规划问题中非常受欢迎。然而，它仍然需要在内存中保存所有生成的节点，对于内存有限或搜索空间极大的应用场景，这就成为了需要解决的问题。 A^* 算法改进有 Dynamic A^* ^[11]、Field D^* ^[12]、Theta * ^[13]、Anytime D^* ^[14]等。

3) PRM 算法

PRM 算法广泛应用于移动机器人的路径规划。该算法通过随机散布一系列点（节点）在配置空间中，然后尝试通过简单的局部规划器连接这些点，构建一个称为路线图的网络，这个网络反映了可以通过的路径。在生成了足够的节点和路线后，用 A^* 算法查找从起点到终点的路径。PRM 算法的主要优势在于其对高维度问题的有效性和对复杂环境结构的适应性，它能够在不同的环境中灵活地重新规划路径，而不需要对整个环境重新进行计算。然而 PRM 算法均匀随

机采样的特性会导致在狭窄区域采样点不足，从而影响路径规划的效果。

为应对上述情形，文献^[15]提出了一种基于高斯分布为样本的采样方法，该方法使样本点的分布依赖于障碍物的密度，由此构建出一个合理的路径地图。为了在狭窄通道内增加采样点，文献^[16]提出了在障碍物的两边抽取点的桥测试法，并以两点中点的空闲性为依据来选取有效采样点，然而这些方法在远离障碍物的区域效率较低。文献^[17]中提出了自适应采样方法，结合概率似然函数和高斯采样，显著提升狭窄区域内的采样点数量，增强 PRM 算法在不确定环境下的适应性。文献^{[11][18]}提出一种改变其采样点的连接点距离的方式，使路径网络图中不合理的路径减少。

综上所述，随着增加探索区域，全局路径规划算法计算代价成倍增长。这意味着在更广阔的区域和更细致的网格中进行路径规划，所需的计算资源和时间将显著增加。Dijkstra 算法是最短路径算法的基础，它适用于权重非负的图，通过贪心策略逐步确定最短路径，但不包含启发式信息，因此在大型图中效率较低。A*算法引入了启发式评估，提高了路径查找的效率，特别是在有明确目标的情况下，但仍然可能在大规模搜索空间中遇到性能瓶颈。PRM 算法则是一种基于采样的运动规划算法，适用于高维度和复杂环境中，通过随机采样和局部路径规划构建一个路线图，这使它在处理动态障碍和复杂环境时具有优势。选择 PRM 算法通常是因为它的高维空间和复杂场景下提供了更好的性能和灵活性，尤其是当环境变化时，它能有效地更新路径而无需从头开始计算。

（2）局部路径规划

1) 模拟退火算法

模拟退火算法作为一种通用概率算法，旨在大范围搜索空间中定位到某一函数的全局最优解，它的灵感来源于固体物理学中的退火过程。在物理学中，退火是指将物质加热至高温，然后慢慢冷却以减少缺陷从而达到低能状态和增加系统稳定性的过程。模拟退火算法模仿这一物理过程，通过引入温度的概念作为控制参数，来模拟求解过程中的“冷却”过程。算法开始时将温度设定为较高值，允许算法在初始阶段可以接受较差的解，以跳出局部最优。随着温度逐渐降低，算法逐渐减少接受劣解的概率，这样做既可以探索广泛的搜索空间，又能确保最终收敛到全局最优解或近似解^[19]。

模拟退火算法的关键在于冷却计划的设置，包括开始温度、结束温度以及温度下降速率，这些因素直接影响到算法的性能和效率。它被广泛应用于优化问题，如旅行商问题、调度问题和各类设计与工程优化问题，因其简单且有效而受到青睐。

2) 人工势场法

人工势场法把移动机器人导航任务比拟于带电粒子在电场中的运动，其中目标点产生吸引势场而障碍物产生斥力势场，移动机器人受到这些势场的组合影响，如同在力场中行走，被吸引到目标点同时绕开障碍物^[20]。

在人工势场法中，目标对移动机器人的吸引力会随着它们之间距离的缩短而逐渐增强，相反，障碍物对移动机器人的排斥力会随着两者间距离的缩短而急剧上升。这种方法的优点是概念简单、实现容易，能够为移动机器人在实时动态环境中生成平滑的路径。但是，它也有可能导致“局部最小”问题，即移动机器人可能陷入无法前往目标且无法离开的区域。为解决这一缺陷，多种改进策略被提出以优化原始的人工势场方法。

3) 时间弹性带算法

时间弹性带算法是一种用于移动机器人和自动驾驶车辆的实时路径规划和避障算法。算法的核心思想是“时空弹性”，它不仅考虑了路径规划的空间元素，还融入了时间作为一个关键维度，这一设计使得时间弹性带算法能够在保持路径平滑性的同时，优化整个移动过程的时间效率。

时间弹性带算法通过对路径中的控制点建立一个弹性带模型来工作，这些控制点可以在保持原路径大致形状的基础上无缝地移动和调整，以应对动态环境中的障碍物和其他变化。弹性带周围会设定一个潜在的代价场，其中包含了障碍物的位置，代价场会根据障碍物的配置导致弹性带的形状发生变化。时间弹性带算法将多个目标函数整合在一个总优化框架里，如路径长度、运动平滑度、动力学和运动学约束、障碍物安全距离等。通过对这些目标函数的权衡和调整，能够计算出一个最优化的时间-空间连续路径。然而在避障过程中，移动机器人需要频繁调整其位姿以达到理想的位置，这不仅会消耗大量时间，还可能使移动机器人陷入“位姿死区”，即任何转向动作都可能导致与障碍物碰撞。这种情况最终可能导致移动机器人的导航任务失败。

4) 基于视觉导航的方法

基于视觉的移动机器人导航是一种利用摄像头作为传感器来获取环境信息并指导移动机器人行动的技术。这一方法通常涉及到场景识别、特征提取、路径决策以及避障等一系列复杂的视觉处理过程。此外，基于视觉的导航系统可以结合如激光雷达、IMU 等传感器来增强稳定性与准确性。由于视觉传感器通常价格较低，易于集成，这使得基于视觉的移动机器人导航越来越受到研究者和工程师的关注，尤其是在自主车辆、无人机和服务移动机器人等领域^[21]。

综上所述，模拟退火算法虽然能高概率找到全局最优解，适用性强，但收敛速度慢，执行时间长；人工势场算法虽然模型直观、实现简单，适用于实时避障和路径规划，但在复杂环境中容易陷入局部最小值，难以处理动态障碍物；时间弹性带算法虽然可以有效避障，但只进行局部优化，无法保证路径的全局最优；基于视觉导航方法获取的图像信息容易受到光照变化、图像模糊以及场景遮挡的影响，导致定位和路径规划不准确，且对计算资源要求较高。

1.2.2 深度学习在移动机器人中的发展及研究现状

大数据时代的到来使数据获取更加容易，加之深度学习技术需要依赖大量数据集进行学习，深度学习已经发展成为当前最热的研究领域之一。在深度学习的多种应用中，目标检测尤为关键，近年来深度卷积神经网络（DCNN）已广泛应用于目标的检测和识别领域^{[22][23]}。在基于深度学习的目标检测算法中，双阶段和单阶段检测算法是最为普遍采用的方法。双阶段方法中的经典算法是 R-CNN^[24]系列算法，是由 Ross Girshich 在 2014 年提出的，支持在向量机 (SVM)^[25]中使用 R-CNN 模型，在候选区域高效地提取特征并将其进行精确的分类，通过建立一个边界盒回归算法，可以计算和测试候选区域的坐标。单阶段方法的典型算法有 YOLO^[26]和 SSD^[27]，Wang 等人提出了 SSD 目标检测算法，他将 YOLO 网络与 Faster R-CNN 网络相结合，并使用非极大抑制(NMS)^[28]将大量重叠的边界框合并成单个边界框，从而把候选区域的确定转化为一个回归问题，并定位最接近总体目标的预测框，从而实现对图像中物体的有效检测。通过这种方式，能够大幅提高计算速度和精度^[29]。YOLO 是一种由 Joseph Redmon 开发的计算机视觉技术，它可以帮助用户快速定位和标记目标，并且可以根据

给定的矩形框来识别出目标所属的类别。YOLO 的出现提高了目标检测的效率和准确性，设计的比大型 NLP transformers 更加小巧，它可以在技术上实现更快的实时推理速率，从而提高效率，并且该算法处理图像非常快，适用于实时处理。

移动机器人在避障方面的发展是机器人技术领域中的一个重要里程碑，它是智能系统与环境交互的基本能力之一。随著传感器技术和计算机视觉的进步，移动机器人现在能够更准确地检测和避开其周围的障碍物，从而在各种环境中安全地导航。

在早期阶段，移动机器人避障主要依赖于超声波和红外传感器，这些传感器可以探测到障碍物的存在，但是对障碍物的形状、大小和具体类型识别能力有限。现代移动机器人采用了更高级的传感器，它们能提供更详细的环境信息，使移动机器人能够创建更精确的地图，并实现更复杂的避障算法。自动驾驶汽车是移动机器人避障技术发展的一个典型例子，它们使用多种传感器来感知周围环境^[30]。通过深度学习算法，自动驾驶汽车能够识别行人、其他车辆、交通信号和道路标记，并在必要时进行快速避障，特斯拉的 Autopilot 和谷歌的 Waymo 是这一领域的突出代表。在工业和商业环境中，仓库内的自动叉车和清洁移动机器人也展示了高级避障技术的应用^[31]，亚马逊使用名为 Kiva 的移动机器人在其仓库中移动货物。这些移动机器人能够识别和规避人类工作人员和其他障碍物，同时执行复杂的货物搬运任务。在更极端的环境中，如火星探测器，避障技术也至关重要^[32]。NASA 的火星漫游车利用板载计算机和摄像头系统进行地表导航，通过识别潜在障碍物，如大岩石或沟壑，来计划安全路径。

未来的发展方向可能包括更高级的人工智能集成，使移动机器人能够更好地理解环境和障碍物的语义信息，以及改进的机器学习算法，以实现更快速和更准确的避障决策。随着技术的不断进步，移动机器人在避障方面的能力将继续提升，使它们能够在更加复杂和不确定的环境中有效地工作。综合来看，深度学习为移动机器人的发展提供了新的可能性，推动了从预定程序到自主智能的转变。随着计算能力的提高和算法的改进，未来的移动机器人将更加智能，更好地融入人类生活和工作环境。

1.3 论文主要研究内容与章节安排

本文研究基于深度学习的移动机器人在真实环境下的导航和避障方法。针对在使用激光 SLAM 进行导航时常常出现移动机器人定位丢失导致地图构建效果不理想的问题,提出融合激光 SLAM、IMU 和里程计数据的方法,以构建更精确的栅格地图,提高移动机器人的定位精度和地图构建的精准度。通过采用改进的轻量化 YOLOv5 模型,提高了实时检测的准确率,使之适合移动机器人的应用需求。针对传统 PRM 算法在狭窄区域时规划路径失败的问题,在学习阶段引入人工势场法,增加狭窄区域的路径网密度,从而提升算法整体性能。研究将深度学习和 TEB 算法相结合用于避障的策略,面对难以避开的障碍物时,移动机器人能够智能地选择其他路径,避免进行频繁的调整操作。使用全局和局部路径规划结合策略,确保移动机器人安全到达目标点。最终,通过采用算法仿真软件和移动机器人平台的实验设施,对提出的方法进行了实验设计,并对移动机器人在室内环境进行 SLAM 的结果进行了深入分析。

论文整体安排如下:

第一章:本章介绍了路径规划的经典算法,概述了深度学习在移动机器人上的发展及应用,并详细说明了各章节的具体内容和结构安排。

第二章:本章简述了移动机器人操作系统和系统模型。为解决传统激光 SLAM 在地图构建中易出现的定位丢失和不准确问题,提出了扩展卡尔曼滤波(Extended Kalman Filter, EKF)结合 IMU 和里程计信息构建地图的方法。

第三章:本章介绍了深度学习的基本概念和网络结构,接着对 YOLOv5 模型进行分析,为了提升检测效果,本章在模型骨干网络中引入 CA 注意力机制,采用 DIoU_NMS 等方法对 YOLOv5 网络模型进行改进。

第四章:本章研究路径规划算法。全局路径规划使用改进的 PRM 算法,局部路径规划采用 TEB 算法与深度学习结合的避障设计,并融合全局和局部路径规划,使移动机器人可以智能导航。

第五章:本章对移动机器人的硬件和软件进行详细介绍,以此搭建一套完整的实车平台。在移动机器人平台上实现深度学习和 ROS 的融合,此外还进行了所使用的 YOLOv5 实时测试,并进行全局路径规划和局部路径规划实验。

最终在移动机器人平台上进行整体对比实验，可从实验分析中得出本文改进算法的实用性。

第六章：总结全文工作，并展望需要进一步研究的内容。

第 2 章 移动机器人系统设计及地图构建

本章介绍了本文使用的移动机器人搭载的 ROS 框架，并介绍了移动机器人坐标系系统模型和阿克曼转向运动学模型。最后，本章基于传统激光 SLAM 技术的地图创建原理，分析其存在的不足，并提出了相应的改进措施。这些改进旨在提升移动机器人实时定位的精度，从而增加地图构建的成功率和提高地图的整体质量。

2.1 ROS 操作系统概述

2.1.1 ROS 发展过程

自 2007 年由斯坦福大学的 STAIR 项目诞生以来，ROS 逐步成长为一个全球性的开源项目，由加州大学伯克利分校的 Personal Robotics Program 与机器人技术公司 Willow Garage 共同推动发展。ROS 的发展过程标志着移动机器人学领域向更加开放和协作的研究环境转变。

ROS 的初衷是为了解决移动机器人研究和开发中存在的重复劳动问题，提供一个共享平台使得研究者和开发者能够共享代码、算法和技术成果，它的设计包括模块化、工具通用性、易于使用和技术透明度等。ROS 通过提供一系列工具和库来支持移动机器人的各个方面，包括但不限于传感器集成、视觉处理、定位和映射、路径规划和机械臂控制。随着时间的推移，ROS 社区迅速扩大并吸引了全球众多研究者、开发者和公司的参与，随着 2010 年 ROS 1 的推出，其成为了移动机器人研究和商业应用的事实标准。ROS 的一个关键特点是其包（Packages）系统，允许用户轻松地共享和分发代码，每个包可能包含 ROS 节点、库、数据集或其他逻辑实体，使得开发复杂移动机器人应用变得更加简单。

ROS 的发展也面临着挑战，特别是在实时性、安全性和可扩展性方面。为了解决这些问题，ROS 2 在 2014 年开始开发，旨在保持 ROS 的核心优势的同时，在通信中间件、安全性和实时性等方面进行改进，ROS 2 的出现标志着对未来移动机器人应用（工业、商业和个人领域）的进一步支持。ROS 从 2010 年

至 2024 年 ROS 发布的所有版本如图 2-1 所示。

ROS Noetic Ninjemys (Recommended)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			May, 2023 (Bionic EOL)
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame	May 23rd, 2016			April, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017
ROS Indigo Igloo	July 22nd, 2014			April, 2019 (Trusty EOL)
ROS Hydro Medusa	September 4th, 2013			May, 2015

图 2-1 ROS 所有版本

2.1.2 ROS 系统特点

今天，ROS 已经成为了移动机器人研究和开发不可或缺的一部分，支持从教育移动机器人到复杂工业自动化系统的广泛应用。它的成功不仅在于技术上的创新，更在于它所代表的开放和协作的社区精神，为机器人学领域的发展做出了巨大贡献。随着技术的不断进步和社区的不断壮大，ROS 将继续在移动机器人技术的未来发展中扮演关键角色。

ROS 的主要特点如下：

(1) 模块化设计：ROS 采用了模块化的设计理念，使得移动机器人系统可以被拆分成多个小的、可管理的部分，这种设计使得开发者可以重用现有的代码来实现新的功能。

(2) 工具生态: ROS 提供了一套丰富的工具集, 包括数据记录与回放、图形化用户界面、仿真环境、可视化工具等, 这些工具支持开发者在设计、测试和调试移动机器人应用时更加高效。

(3) 通信机制: ROS 内部的通信机制包括话题 (Topics)、服务 (Services) 和动作 (Actions), 支持不同节点间的数据交换和通信。这种灵活的通信框架允许移动机器人系统中各个部件之间进行高效的信息交流和协作。

(4) 社区支持和开源文化: ROS 的一个显著特点是其强大的社区支持和开源文化。全球的研究者和开发者贡献了大量的教程、文档、论坛讨论和代码库, 使得新手可以快速入门, 专家也能在复杂的问题上寻求帮助。

(5) 跨平台和多语言支持: ROS 支持多种操作系统, 如 Ubuntu、MacOS、Windows, 同时提供了 C++、Python、Lisp 等多种编程语言的接口, 满足不同开发者的需求。

(6) 可扩展性: ROS 的架构设计具有高度的可扩展性, 可以适用于从小型教育移动机器人到大型工业移动机器人系统的开发。通过 ROS, 开发者可以轻松集成新的硬件设备, 扩展系统的功能。

综上所述, ROS 以其绝对的优势在全球机器人领域内建立了不可替代的地位, 为移动机器人研究与开发提供了强有力的支持。

2.2 移动机器人系统模型

2.2.1 坐标系系统模型

移动机器人涉及三种主要的坐标系: 移动机器人本体坐标系、传感器坐标系以及全局坐标系。移动机器人的本体坐标系一般设定于移动机器人中心或重心, 传感器坐标系则通过固定的偏移量与移动机器人本体坐标系关联, 全局坐标系则描述了移动机器人周围环境的空间布局。图 2-2 所示, 移动机器人在平面二维坐标系下的位置, 全局坐标系 xoy , 移动机器人本体坐标系 XOY , 传感器坐标系 $X_1O_1Y_1$, 传感器坐标系在 X 轴和 Y 轴上方向一致, 在 Y 轴上差一个偏移量。 θ 为全局坐标系下 x 轴到移动机器人坐标系 X 轴的角度, 在某一时刻的位姿用 (x, y, θ) 表示。

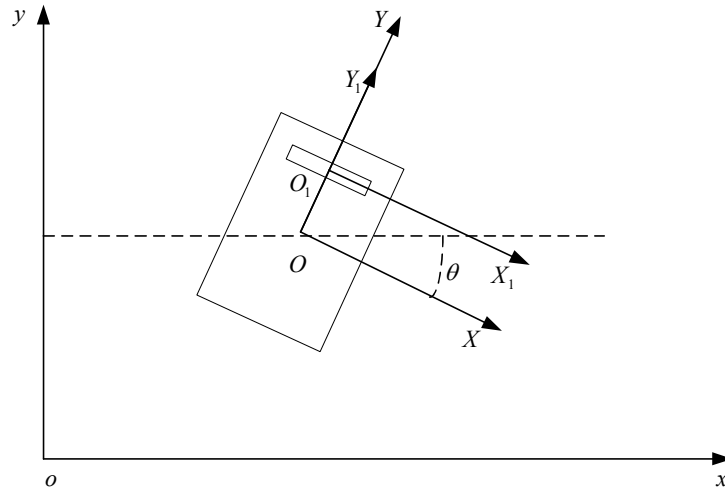


图 2-2 移动机器人坐标系系统

移动机器人坐标系转换全局坐标系变换矩阵：

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-1)$$

全局坐标系转换移动机器人坐标系变换矩阵：

$$R(\theta)^{-1} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-2)$$

2.2.2 阿克曼转向运动学模型

本文移动机器人平台使用了阿克曼转向机制，并以此构建移动机器人的运动模型，阿克曼转向模型如图 2-3 所示。

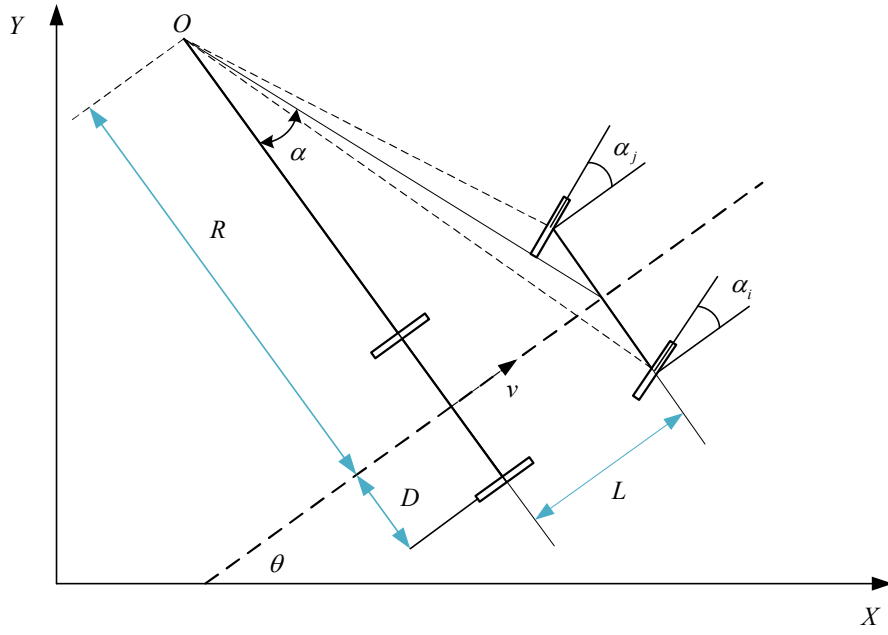


图 2-3 阿克曼转向模型

其中 θ 为移动机器人在世界坐标系下的朝向， α 为移动机器人的转向角， α_i, α_j 分别为移动机器人前轮外侧轮胎转向角和内侧轮胎转向角， O 为移动机器人转弯圆心， R 为移动机器人转弯半径， L 为移动机器人轴距， $2D$ 为移动机器人轮距，

由于阿克曼转向模型的内外轮胎转向角不同，因此需要统一的移动机器人转向角，移动机器人转弯半径 R 如式（2-3）。

$$R = \frac{L}{\tan \alpha} \quad (2-3)$$

内外侧轮胎角度 α_i, α_j 如式（2-4）所示。

$$\begin{aligned} \alpha_i &= \arctan \frac{L}{R+D} \\ \alpha_j &= \arctan \frac{L}{R-D} \end{aligned} \quad (2-4)$$

当移动机器人以速度 v 、转向角 α 进行转弯时，可以得到内、外侧后轮的速度如式（2-5）所示。

$$\begin{aligned} v_i &= v \frac{R+D}{R} \\ v_j &= v \frac{R-D}{R} \end{aligned} \quad (2-5)$$

设后轮轮轴中点在世界坐标系下的坐标为 (x, y) ，可以得到整体的运动学模型如式（2-6）所示。

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= v \frac{\tan \alpha}{L}\end{aligned}\quad (2-6)$$

2.3 激光 SLAM 地图构建

2.3.1 传统激光 SLAM

SLAM 技术旨在解决移动机器人在未知环境下运动时的定位与地图构建问题^[33]。基于视觉解决定位和建图问题的 SLAM 称为 Visual SLAM^[34]，这种方法成本较低，易受环境光线变化的影响。而激光 SLAM 利用激光雷达进行环境扫描，不易受光照条件影响，能快速获取大量且种类丰富的数据，构建的三维地图精度也相对较高^[35]。激光 SLAM 系统通常配备多种传感器，由于激光雷达精度高，能够在平面环境中径向精确扫描，因此在对地图精度要求高的应用场景中，激光 SLAM 是首选方案。

由于激光雷达能够生成高精度的地图，它通常被用在对地图精确度有较高要求的地图构建任务中^[36]。在进行环境的平面扫描时，激光雷达沿着径向方向捕获图像，以便准确计算出扫描点的坐标位置。

传统的激光 SLAM 技术核心依托于激光雷达传感器构建地图，而在某些情况下也会融合额外的传感器来优化地图的生成过程。本章基于滤波的激光 SLAM，使用 Gapping 算法通过粒子滤波方法进行地图构建，它使用粒子滤波器来维护多个粒子，每个粒子代表地图的一个可能实例及移动机器人的位置。其主要架构展示在图 2-4 中。在 ROS 环境下，该算法主要通过 Gapping 软件包实现，通过搜集激光扫描与里程计数据，借助 Openslam Gmapping 包对数据进行初步处理，以此来完成地图的创建。

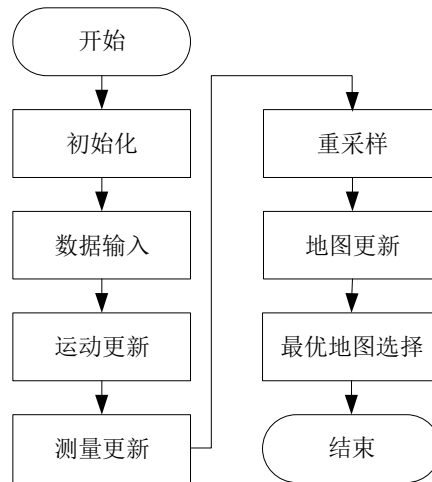


图 2-4 Gmapping 算法实现框架

1、初始化：Gmapping 始于移动机器人和地图的初始状态。它初始化一系列粒子，每个粒子包括移动机器人的一个假设位置和空白地图。

2、数据输入：当机器人移动时，激光雷达传感器收集数据，并将这些数据与移动机器人的运动信息一起输入到算法中。

3、运动更新：每当机器人移动时，算法更新所有粒子的移动机器人位置，根据移动机器人的运动模型预测移动机器人的新位置。

4、测量更新：通过比较激光雷达数据与粒子表示的地图，算法评估每个粒子的质量。雷达扫描与粒子地图的匹配程度越高，粒子的权重越大。

5、重采样：为了避免粒子退化问题，Gmapping 通过重采样步骤来选择一组新粒子高，权重的粒子会被复制，低权重的可能被舍弃。

6、地图更新：随着新的激光数据的到来，算法会更新粒子对应的地图，这些更新可以增加地图中的特征，如墙壁和障碍物。

7、最优地图选择：从所有粒子代表的地图中，选择最有可能的地图作为当前的最佳估计。

8、输出：最终，算法输出一个优化后的地图和移动机器人的估计轨迹。

整个过程是迭代的，随着移动机器人的持续移动和环境探索，地图逐渐变得更加完整和精确。Gmapping 算法的优势在于其对环境的适应性和在计算上相对效率，但它也依赖于精确的运动模型和激光雷达的高质量数据，且对于动态环境和大规模地图构建可能会存在局限性。

2.3.2 传统激光 SLAM 缺陷

本文移动机器人平台搭载了 Ydlidar X4 激光雷达，参数如表 2-1 所示。

表 2-1 Ydlidar X4 单线高精度激光雷达参数

参数项	数值
测距频率	5000 Hz
扫描频率	6-12 Hz
测距范围	10 m
扫描角度	0-360 Deg
俯仰角	0.25-1.75 Deg

在采用传统的激光 SLAM 方法进行地图构建时，常常会遇到自定位漂移的问题，从而影响了所得到的地图精度，如图 2-5 所示：

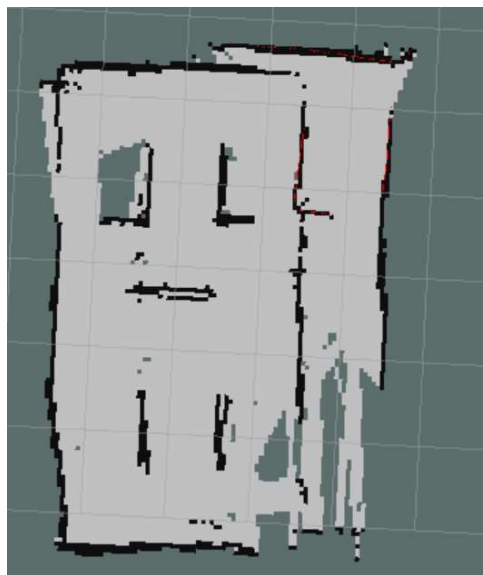


图 2-5 激光雷达建图

激光 SLAM 技术使用激光雷达作为主要传感器，它能够精确测量周围环境中对象的距离和形状，构建出高精度环境地图。在使用激光 SLAM 进行地图构建时，移动机器人快速的移动和计算上的延迟往往会造成定位的不精确和位姿的偏移，进而影响地图的准确性。同时，由于激光雷达自身并不能直接提供移动机器人的精确位姿信息，仅能通过后续计算来估算，这种方式难免随时间积累计算误差，增加激光 SLAM 建图失败的风险。

激光雷达导航的移动机器人主要依赖其传感器系统来感知周围环境，但是仅靠单一传感器所提供的信息可能不够准确和可靠。通过多传感器融合和补偿

技术，可以有效地结合不同传感器的数据，利用它们来提高环境感知的精度、速度和稳定性。

为了抑制移动机器人运动过程中的误差累积，文献^[37]提出了多传感器组合导航的方案，该方案通过使用其他类型的传感器来补偿激光雷达定位中的误差，从而提高整体定位的准确度。此外，针对仅依赖激光雷达造成的定位精度低、高端激光雷达昂贵等问题，文献^[38]对 AGV 的定位和导航进行了研究，采纳了激光雷达结合视觉定位的方法，这种结合方式不仅提高了定位的精度，还增加了路径的灵活性和运行的智能化，展现了综合使用多种传感器在提升导航性能方面的优势。

2.4 EKF 融合里程计与 IMU 数据建图

本文用 EKF 融合激光雷达，里程计和 IMU 数据结合来增加定位精度^[39]。EKF 作为一种公认的线性最优估计方法，它可以对多传感器数据进行优化估计，其利用状态空间模型来连续追踪和纠正来自不同传感器的误差，并融合这些数据以获得更加准确的估计结果^[40]。

2.4.1 IMU 原理

惯性测量单元（IMU）是一种电子设备，用于测量一个物体的特定物理量，包括三轴方向的角速度和加速度，有时还包括倾斜角和磁场向量^[41]。IMU 通常由陀螺仪和加速度计组成，可以提供关于物体运动的关键信息，通过对 IMU 数据的处理和分析，可以实现物体的姿态估计、定位和导航等功能。尽管 IMU 能够提供实时的运动信息，但其数据可能受到噪声的影响，并可能随时间积累误差。

IMU 能够实时捕获移动机器人的加速度信息，这包括三轴的线性加速度和三轴的角加速度共计六种加速度数据，对加速度数据二次积分，能得到移动机器人的位姿信息。移动机器人在空间中的移动可被视为矩阵变换，即通过矩阵乘法来实现变换，将坐标原点设为移动机器人的初始状态，以该点分别绕 X 、 Y 、 Z 轴旋转可以表示为 x 、 y 、 z 角度，则位置的估算如式(2-7)、(2-8)所示。

$$O = \begin{bmatrix} C_y & 0 & -S_y \\ 0 & 1 & 0 \\ S_y & 0 & C_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_x & S_x \\ 0 & -S_x & C_x \end{bmatrix} \begin{bmatrix} C_z & -S_z & 0 \\ -S_z & S_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-7)$$

$$O = \begin{bmatrix} C_z C_y - S_x S_y S_z & C_y S_z + C_z S_x S_y & -C_x S_y \\ -C_x S_z & C_z C_x & S_x \\ C_z S_y + C_y S_z S_x & S_z S_y - C_z C_y S_x & C_x C_y \end{bmatrix} \quad (2-8)$$

移动机器人位姿解算本质上是利用位姿传感器实时估计移动机器人的姿态矩阵 O ，以此来确定具体位置。

2.4.2 EKF 数据融合

移动机器人的位姿信息可以通过使用 EKF 算法进行估算，其中位姿融合状态模型是基于里程计和 IMU 数据建立的，这两种传感器提供了速度和转向角的信息^[42]。系统状态转移以及观测方程为：

$$\begin{cases} x_{t+1} = g(x_t, u_t) + \varepsilon_{t+1} \\ z_{t+1} = h(x_{t+1}) + \delta_{t+1} \end{cases} \quad (2-9)$$

预测模型为：

$$\begin{cases} \hat{x}_{t+1} = f(\hat{x}_t, u_t) \\ P_{t+1} = F_t P_t F_t^T + Q_t \end{cases} \quad (2-10)$$

更新模型为：

$$\begin{cases} K_{t+1} = P_{t+1}^- H_t^T (H_t^T P_{t+1}^- H_t^T + R_{t+1})^{-1} \\ \hat{x}_{t+1} = \hat{x}_{t+1}^- + K_{t+1} (z_{t+1} - h(\hat{x}_{t+1}^-)) \\ P_{t+1} = (I - K_{t+1} H_t) P_{t+1}^- \end{cases} \quad (2-11)$$

其中： $F = f'(\hat{x}_t)$ ， $H = h'(\hat{x}_{t+1})$ 。

移动机器人 t 时刻的位姿为：

$$x_t = (X_t, Y_t, \theta_t, v_x^t, v_y^t, w_t)^T \quad (2-12)$$

其中， (X_t, Y_t, θ_t) 表示位姿信息， (v_x^t, v_y^t, w_t) 表示速度信息。可以得出状态转移方程为：

$$x_{t+1} = x_t + \begin{pmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} (v_x^t, v_y^t, w_t)^T \quad (2-13)$$

其中, $(v_x^t, v_y^t, w_t)^T$ 表示控制输入。所以移动机器人里程计数据观测方程如式 (2-14) 所示。

$$z_{odom,t} = H_{odom} x_t = I_6 (X_t, Y_t, \theta_t, v_x^t, v_y^t, w_t)^T + e_{odom,t}(d) \quad (2-14)$$

其中, H_{odom} 和 I_6 表示轮式里程计的预测矩阵, $e_{odom,t}(d)$ 表示噪声协方差矩阵。

由于本文的移动机器人系统仅研究二维环境地图, 因此不需要 IMU 数据中 X 轴和 Y 轴的数值。所以移动机器人 IMU 数据观测方程如式 (2-15) 所示。

$$z_{imu,t} = H_{imu} x_t = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} (X_t, Y_t, \theta_t, v_x^t, v_y^t, w_t)^T + e_{imu,t} \quad (2-15)$$

其中, H_{imu} 表示 IMU 的预测矩阵, $e_{imu,t}$ 表示噪声协方差矩阵。

结合式 (2-14) 和 (2-15) 可得出最终的系统预测方程:

$$z_t = \begin{pmatrix} z_{odom,t} \\ z_{imu,t} \end{pmatrix} = \begin{pmatrix} H_{odom} \\ H_{imu} \end{pmatrix} x_t + \begin{pmatrix} e_{odom,t}(d) \\ e_{imu,t} \end{pmatrix} = H_t x_t + e_t(d) \quad (2-16)$$

其中系统观测转移矩阵 H_t 为:

$$H_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (2-17)$$

系统观测噪声协方差矩阵 $e_t(d)$ 为:

$$e_t(d) = \begin{pmatrix} e_{odom,t}(d) & 0 \\ 0 & e_{imu,t} \end{pmatrix} \quad (2-18)$$

将上述公式导入 (2-10)、(2-11) 可得出移动机器人最优位姿估计。

数据融合流程如图 2-6 所示。

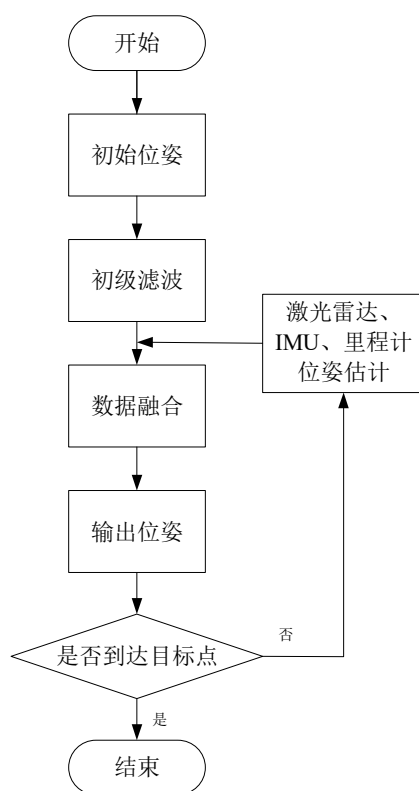


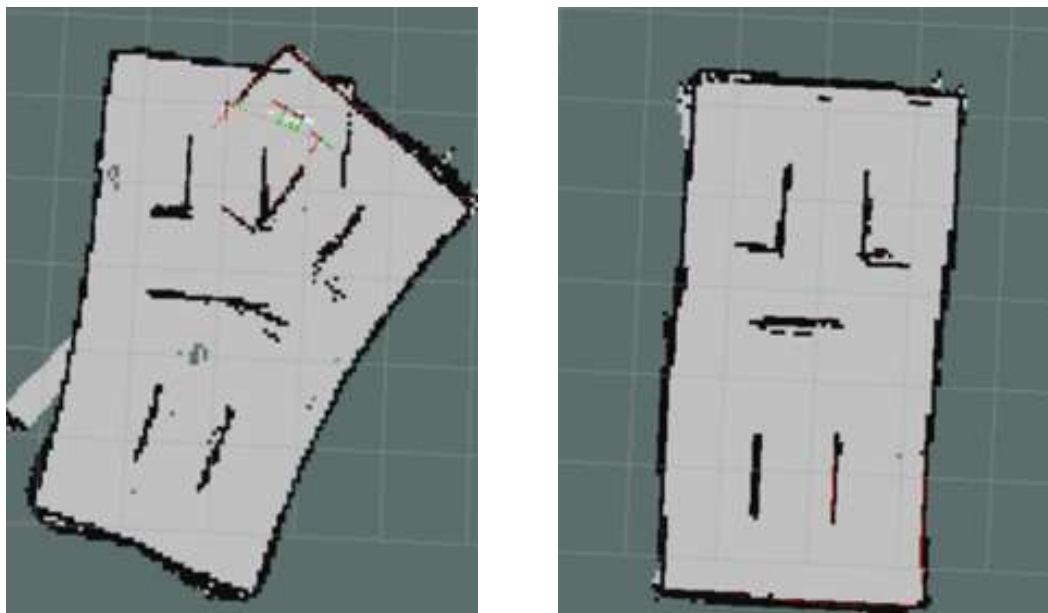
图 2-6 数据融合流程图

在构建栅格地图的过程中，本文获取了 IMU 与里程计的数据，随后通过数据筛选提取速度与位姿信息，如图 2-7 所示。

```
learningx@learningx: ~
learningx@learningx:~$ rostopic echo /raw/data_raw
header:
  seq: 3451
  stamp:
    secs: 1613622296
    nsecs: 840082660
  frame_id: "imu_link"
orientation:
  x: 0.0
  y: 0.0
  z: 0.0
  w: 0.0
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: -0.000106414676712
  y: -0.000989656493424
  z: -0.000404375771506
angular_velocity_covariance: [0.04, 0.0, 0.0, 0.0, 0.04, 0.0, 0.0, 0.0, 0.04]
linear_acceleration:
  x: 1.68138977044
  y: 0.162697769122
  z: 4.56583191546
linear_acceleration_covariance: [0.04, 0.0, 0.0, 0.0, 0.04, 0.0, 0.0, 0.0, 0.04]
---
```

图 2-7 位姿及速度信息

实验分为仅使用激光 SLAM 进行地图构建和将激光 SLAM、IMU 和里程计数据融合进行地图构建，建图结果如图 2-8 所示。由图可看出，激光 SLAM、IMU 和里程计数据融合后所构建的地图没有发生偏移。



(a) 激光SLAM建图

(b) 激光SLAM、IMU、里程计数据融合建图

图 2-8 建图结果对比

2.5 本章小结

本章首先对移动机器人用到的 ROS 操作系统进行介绍，明确移动机器人构建地图与路径规划的功能需求。然后介绍了坐标系系统模型和阿克曼转向运动学模型。最后采用了 EKF 算法来融合 IMU 和里程计数据，有效提升了移动机器人定位的准确性，从而增加了地图构建成功成功率。

第3章 基于深度学习的目标检测

本章介绍了深度学习中卷积神经网络结构和 Darknet 深度学习框架。以 YOLOv5 为基础网络模型进行改进，加入 CA 注意力机制、引入递归门控卷积等操作，提升目标检测的精度。在公共数据集上与基础网络模型和其他网络模型进行对比，改进后的算法检测效果显著提升，拥有较好的实用性。

3.1 卷积神经网络概述

卷积神经网络是一种深度学习模型，专门用于处理网格拓扑结构的数据，它在图像识别、视频内容分析以及自然语言处理等多个领域得到广泛应用，因其强大的特征提取能力而受到青睐^[43]。

3.1.1 基本网络结构

卷积神经网络是一种前向传播的神经网络，卷积操作本质上是两个函数重叠部分的积分表达。深度学习算法模型能高效地从图像数据中抽取重要信息，并进行特征的辨识与归类^[44]。

1) 卷积层

卷积层主要用于特征提取，通过一组可学习的滤波器（或核）来扫描输入图像或前一层的特征图，从而捕捉局部依赖和图像特征。每个滤波器负责检测输入数据中的特定低级特征^[45]，如边缘或颜色斑块。通过在不同的空间位置应用相同的滤波器，卷积层保持了对空间层级结构的感知，减少了模型参数的数量，并提高了模型处理图像变化的能力。卷积过程如图 3-1 所示。

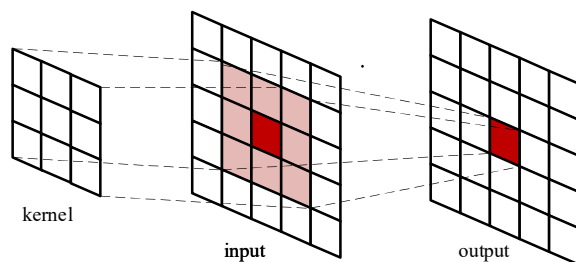


图 3-1 卷积过程

卷积核也被称作卷积矩阵或卷积掩码，是一种比卷积操作对象图像小的小型矩阵。卷积操作通过卷积核在输入数据上移动来实现。在每个位置，卷积核与输入数据的对应元素相乘后求和，生成新的特征图。这使得网络能够捕捉到输入数据的局部模式，如边缘或纹理。通过这种方式，卷积操作有效地减少了模型的参数数量，加强了模型对图像平移的不变性，为深度学习模型提取图像等数据的关键特征提供了强大的能力。

如图 3-2 所示，这里的过滤器是一个 3×3 矩阵，元素为 $[2, 0, 1]$, $[1, 2, 0]$, $[0, 1, 2]$ 。滤波器在输入端滑动输入图像的大小为 5×5 ，内核的大小为 3×3 ，元素为 $[2, 0, 1]$, $[1, 2, 0]$, $[0, 1, 2]$ ，步幅为 1。

3 ₂	3 ₀	2 ₁	1	0
0 ₁	0 ₂	1 ₀	3	1
3 ₀	1 ₁	2 ₂	2	3
2	0	0	2	2
2	0	0	0	1

13	15	19
7	12	15
10	4	13

图 3-2 卷积计算

2) 池化层

池化层主要用于降低特征图空间尺寸，减少网络中参数数量，缩短计算时间，并帮助减轻过拟合现象^[46]。它通过对邻近像素进行汇总操作来实现这一点，常见的汇总方法包括最大池化和平均池化^[47]。池化层通过这种下采样方法，保持了特征的代表性，同时提高了模型对输入变化的鲁棒性。

最大池化取区域内的最大值作为输出，最大池化如图 3-3 所示。平均池化就是计算区域内所有像素的平均值，平均池化如图 3-4 所示。

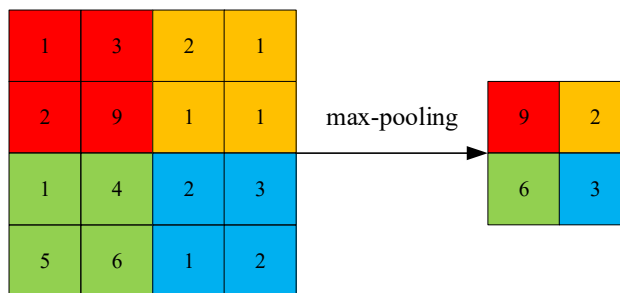


图 3-3 最大池化

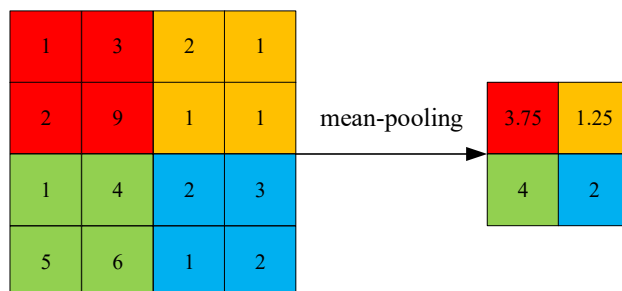


图 3-4 平均池化

3) 激活函数

激活函数在人工神经网络中发挥着至关重要的作用，它决定了一个神经元是否应该被激活，从而能够对输入的信息做出有意义的响应。激活函数引入非线性因素，使得神经网络能够学习和执行更加复杂的任务，如分类和回归。没有激活函数，神经网络无论复杂度如何，都只能表达线性关系，常用的激活函数包括 Sigmoid、ReLU 等。

(1) Sigmoid 函数

Sigmoid 函数是一个广泛使用的激活函数，形状像一个 S 型曲线。它将任意实数值映射到(0,1)区间，具有平滑渐变和输出范围限定的特点，常用于二分类问题中。函数如式(3-1)，函数图像如图 3-5 所示。

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3-1)$$

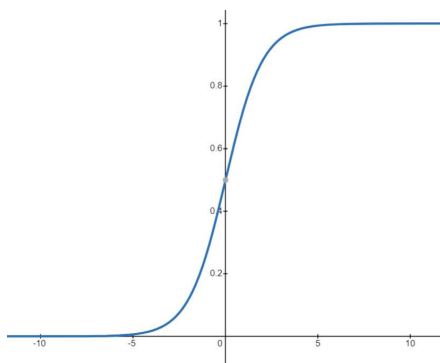


图 3-5 Sigmoid 函数

(2) ReLU 函数

ReLU 函数是一种简单的非线性激活函数，对于输入的正值直接输出该值，而对于负值则输出零。函数如式(3-2)，函数图像如图 3-6 所示。

$$\text{ReLU}(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (3-2)$$

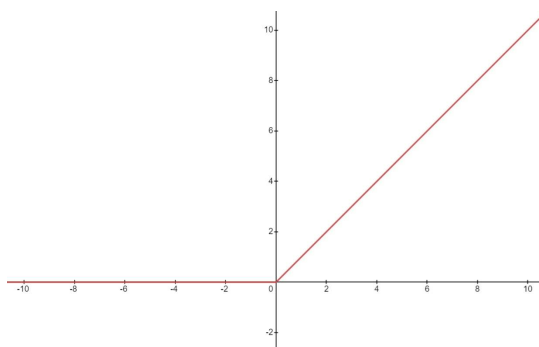


图 3-6 ReLU 函数

4) 损失函数

损失函数是用于衡量模型预测结果与真实值之间差异的指标。它是一个非负值，表示模型性能的“损失”程度，其中损失越低，模型的预测越准确。常见的损失函数包括均方误差用于回归问题，交叉熵损失用于分类问题，以及更复杂的函数，如 Hinge 损失和 Huber 损失等。

(1) 均方差损失函数

均方差损失函数是一种常用的回归分析方法，它是通过对模型预测值和实际值之差的平方的平均来衡量模型的性能。其优点是易于理解和计算，但受异常值影响较大，均方差损失函数如下：

$$J(\theta) = \frac{\sum_{i=1}^n (y^{(i)} - y^{(i)'})^2}{n} \quad (3-3)$$

其中 $y^{(i)}$ 是实际值， $y^{(i)'}$ 是预测值， n 是样本数量。

(2) 交叉熵损失函数

在分类问题中，通常采用交叉熵损失函数来评价模型的预测精度。该方法通过测量实际数据与模型预测结果的差值来表示。交叉熵损失函数如下：

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \quad (3-4)$$

其中 y 表示真实的分布， $h(x)$ 为训练后的预测分布。

3.1.2 Darknet 深度学习框架

深度学习框架的出现极大地简化了深度学习技术应用到实际中的过程，深度学习框架有 TensorFlow^[48]、Keras^[49]、Torch^[50]、Caffe^{[1][51]}、Darknet^[52]、Mxnet^[53]等。Darknet 是一个开源的深度学习框架，由 Joseph Redmon 开发，用 C 和 CUDA 编写，专为神经网络研究而设计，被广泛用于计算机视觉任务。它以其轻量级和高性能而知名，尤其适合在中等规模的硬件上进行实时目标检测。

Darknet 框架各层次的主要功能如下：

应用层：在 Darknet 框架中，可视为构建在其核心之上的各种应用，如图像识别、目标检测等，这一层包括了使用 Darknet 为特定任务编写的高级脚本或程序。

训练执行层：对应于训练模型的过程。此层负责接收训练指令，执行训练循环，调整模型权重并优化性能，它直接调用下层提供的功能，如反向传播和权重更新算法。

功能实现层：此层包括了 Darknet 中实现的各种深度学习算法和网络结构。例如，卷积层、池化层、全连接层等网络基础组件的实现，以及对目标检测等具体任务的算法实现。

核心计算层：是 Darknet 的数学引擎，处理所有低级数学运算，例如卷积运算，激活函数的应用，梯度的计算等。此层是算法执行的基础，优化了计算过程以利用 CPU 和 GPU 的性能。

I/O 层：负责数据的输入与输出管理，包括读取训练和测试数据集，解析网络配置文件，以及保存或加载预训练模型等功能。

数据结构定义层：提供了框架中利用的各种数据结构的定义，包括网络结构、层结构、权重矩阵等。这一层是框架其他组件工作的数据基础。

Darknet 框架图及源码结构划分如图 3-7 所示。



图 3-7 Darknet 框架图及源码结构划分

3.2 深度学习网络结构设计

3.2.1 YOLOv5 网络模型

YOLO 算法以其速度快和准确性高而著称，在单次前向传播中就能实现图像中多个物体的实时检测。YOLOv5 优化了网络结构，减少了模型大小并提高了性能。它使用卷积神经网络来提取特征，并利用先验框进行对象检测，广泛应用于计算机视觉领域，包括自动驾驶、监控系统等。本章选择的 YOLOv5s 网络是 YOLOv5 的轻量级版本，YOLOv5 的网络结构如图 3-8 所示。输入端、骨干网络、颈部网络和输出端是 YOLOv5 的 4 个部分。

1) 输入端

YOLOv5 的输入端使用了 Mosaic 数据增强，它运用随机排布、缩放和裁剪的方式将 4 张图片拼接作为输入，从而填充照片的背景，同样也有助于增加批量大小，而且在完成批量归一化时会计算 4 张照片，因此不太依赖本身批量大

小，减少 GPU 内存损耗，加快网络训练速度。

2) Backbone 网络

YOLOv5 的主干网络采用 CSPDarknet53，包括 Focus 结构、CSPNet 结构、SPP 模块等，Focus 结构是一种在数据端进行数据增强的方法。在处理较大的特征图时，YOLOv5 可以将输入的数据切分为 4 份，然后在 channel 维度拼接，然后传递连接层以堆叠图像，以便通过卷积层提取不同级别的特征表示。CSPNet 框架构成骨干网络，通过残差结构连接增强不同维度特征图的特征融合能力。SPP 模块在四个不同维度进行最大池化，然后进行 Concat 合并组合，最终将输出恢复到同初始输入一致。

3) Neck 网络

YOLOv5 中的 Neck 网络采用 PANet 产生一个特征金字塔网络，然后把这些特征融合到输出中，并将其传递给输出端做预测。该方法采用了特征金字塔网络（FPN）和路径聚合网络（PAN）两种不同的结构，FPN 可以将语义信息从深层特征图像传递到浅层特征图像。而 PAN 可以将位置信息从浅层特征层传输到深层特征层。通过将 FPN 和 PAN 的相结合，可以将多个中继层的参数进行聚合，大大增强了网络的特征融合能力。

4) 输出端

输出端由损失函数和非最大值抑制组成，YOLOv5 的损失函数包括分类损失、定位损失和置信度损失。YOLOv5 用 CIoU loss 计算定位损失，用 BCE loss 计算置信度损失与分类损失。NMS 主要用于去除多余的检测框，保留预测概率最高的候选框作为最终预测框。

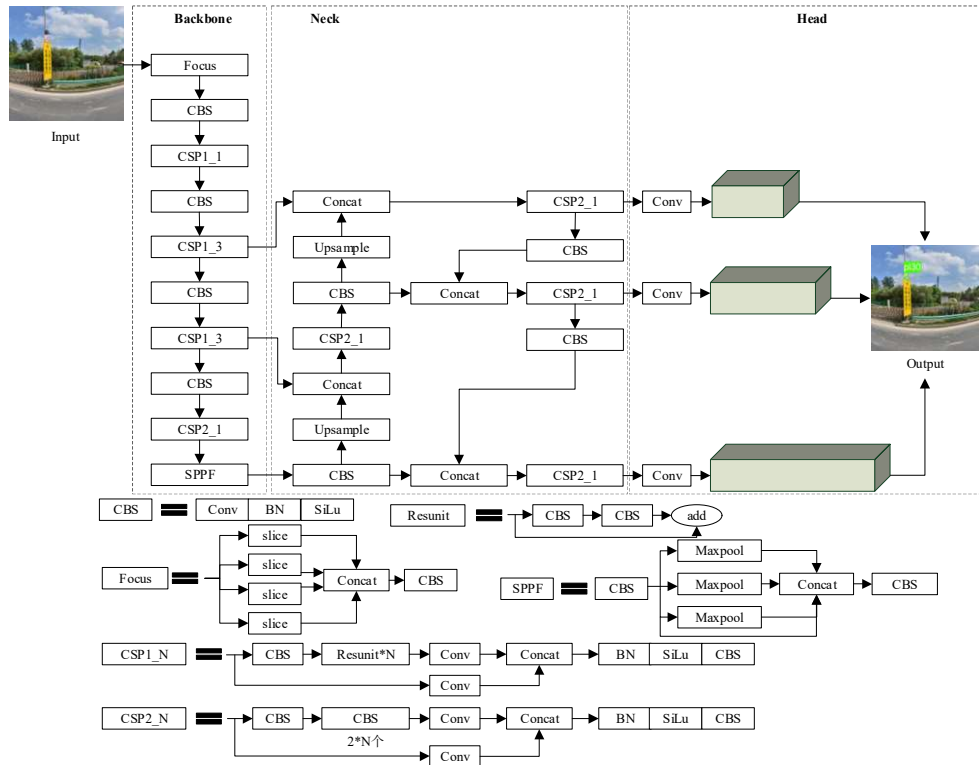


图 3-8 YOLOv5 网络结构

3.2.2 YOLOv5 模型的改进及优化

本章以 YOLOv5 网络模型为基础，首先在骨干网络中加入 CA 注意力机制，考虑了通道信息和方向相关的位置信息，提高检测精度；然后引入了 DIoU_NMS，它将中心点考虑进来保留更多的矩形框，提高遮掩重叠交通标志的识别精度；其次加入了递归门控卷积($g^n\text{Conv}$)，实现关键特征之间的高阶交互；最后，在三个输出检测的基础上增加了一个小目标检测头，增强小目标检测的性能。

1) 引入注意力机制 CA

由于深度学习技术的飞速发展，注意力机制已经成为自然处理、计算教学、图象测试、语音识别等应用领域的关键技术。它能够根据不同的情况，以高权重关注主要数据，以低权重忽视不相关的数据，而且能够根据需要不断地调节权重，以便更好地判断出重要的数据。常见的注意力机制有：CBAM (Convolutional Block Attention Module)、SENet (Squeeze-and-Excitation Networks) 和 CA 等。CBAM 是一种轻量化的、包含通道和空间两种注意机制

的卷积注意模块，增加了一个并行的最大池化层，引入了大尺度的卷积核提取空间特征，但忽略了长程依赖问题。SENet 模块是一个即插即用的模块，特征图通过自适应全局平均池化，压缩特征层长宽，只留下通道维度的信息。然后连续使用两个全连接层对通道信息进行自注意。最后，将全连接输出通过一个 sigmoid 函数，得到 0-1 的通道权重，最后对原特征层进行加权，生成加权后的特征图。然而，目前的研究主要集中在通道间的相关性上，而忽视了通道的空间特性。

为解决以上问题，本文引用了 CA 注意力机制，它不仅考虑了通道信息，还考虑了方向相关的位置信息，并且足够的灵活和轻量，能够简单的插入到轻量级网络的核心模块中。CA 注意力机制算法流程图如图 3-9 所示：

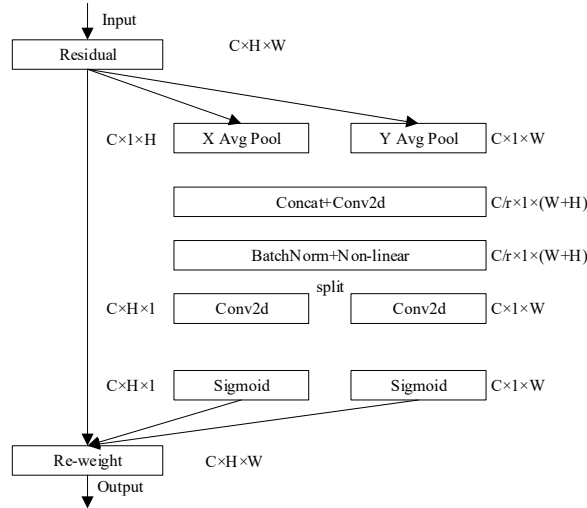


图 3-9 CA 注意力机制算法流程图

为了避免空间信息全部压缩到通道中，CA 注意力机制没有使用全局平均池化，而是分解了全局平均池化，首先利用两个一维池化核分别沿着水平和垂直坐标编码所有通道。

第 c 通道中高度 h 的输出，公式如下：

$$z_c^h(h) = \frac{1}{W} \sum_{0 \leq i \leq W} x_c(h, i) \quad (3-5)$$

第 c 通道中宽度 w 的输出，公式如下：

$$z_c^w(w) = \frac{1}{H} \sum_{0 \leq j \leq H} x_c(j, w) \quad (3-6)$$

其次，拼接上式生成的两个带有方向感知的特征图，公式如下：

$$f = \delta(F_1([z^h, z^w])) \quad (3-7)$$

然后, 沿着空间维度, 再将拼接后的特征图进行 split 操作, 分别利用 1×1 卷积进行升维度操作, 再结合 sigmoid 激活函数分别得到最后的坐标注意力, 公式如下:

$$g^h = \sigma(F_h(f^h)) \quad (3-8)$$

$$g^w = \sigma(F_w(f^w)) \quad (3-9)$$

最后, 输出为原输入与两个坐标注意力相乘, 公式如下:

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \quad (3-10)$$

CA 注意力机制可以更加精准地定位和识别感兴趣的目标, 还可以在轻量级网络 YOLOv5s 的基础上给下游任务带来巨大的增益。本文中, 在 Backbone 中的 C3 中加入了 CA 注意力机制。

2) 引入递归门控卷积

递归门控卷积用门控卷积和递归设计实现高阶空间交互, 是用标准卷积、线性投影和元素乘法构建的, 但有与自注意类似的输入自适应空间混合功能。

设输入特征 $x \in \mathbb{R}^{HW \times C}$, 则 $g^n \text{Conv}$ 的输出如下:

$$\begin{aligned} [p_0^{HW \times C}, q_0^{HW \times C}] &= \phi_{in}(x) \in \mathbb{R}^{HW \times 2C}, \\ p_1 &= f(q_0) \odot p_0 \in \mathbb{R}^{HW \times C}, \\ y &= \phi_{out}(p_1) \in \mathbb{R}^{HW \times C}, \end{aligned} \quad (3-11)$$

其中, ϕ_{in} 和 ϕ_{out} 为执行通道混合的投影层, f 为深度卷积, $\odot$ 为点积操作。

循环执行门控卷积运算, 并通过引入更高层次的相互作用, 进一步增加了模型的计算能力。首先利用 ϕ_{in} 提取出一系列的投影特征, 再对其进行递归的执行门控卷积; 其次, 按照 $1/\alpha$ 比例调整输出, 使训练更加稳定, 并且按不同次序匹配维度; 最后, 向投影层 ϕ_{out} 输入最终递归步骤 q_n 的输出, 得到 $g^n \text{Conv}$ 的结果, HorNer 基本构建块如图 3-10 所示。

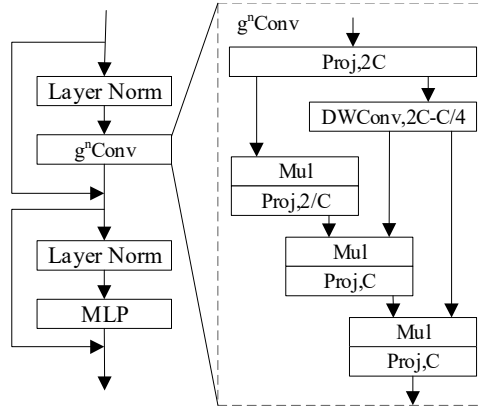


图 3-10 HorNet 基本构建块

$g^n\text{Conv}$ 是一种非常灵活的、可定制的卷积块，可以兼容多种卷积变体，并且可以把二阶交互推广到任意阶，无需增加附加的运算。本文在 Head 模块的 CSP 中加入 HorNet 基本构建块，具体如图 3-11 所示。

3) NMS 改进

NMS 即非极大值抑制，在 YOLOv5 算法中，最终在一幅图像中会有多个疑似目标的矩形框，并为每个矩形框标出类别的概率，然后用 NMS 删除准确率低的矩形框，保留准确度最高的矩形框。

YOLOv5 使用的是传统 Greedy_NMS 方法，它以分类置信度为基础，对其进行排序，然后选择置信度最高的一个矩形框作为参考框，再结合其他的矩形框来计算 IoU，将得到的 IoU 数值超过了设置的阈值的矩形框剔除，而低于的则保留，然后重复此过程，直到出现准确最高的矩形框。NMS 算法如下：

$$s_i = \begin{cases} s_i, & DIoU(M, b_i) < \varepsilon \\ 0, & DIoU(M, b_i) \geq \varepsilon \end{cases} \quad (3-12)$$

其中： s_i 为分类得分， M 为置信度最高的矩形框， b_i 为小于阈值的其他矩形框， ε 为设定的阈值。

但是在大多数的情况下，IoU 与置信度没有太大的相关性，因此许多置信度高的矩形框的位置不是很准确，YOLOv5 加上传统 NMS 在识别重叠区域时，更有可能出现遗漏和错误。为了提高模型对目标数量多、遮挡物体和重叠区域的检测性能，用 DIoU_NMS 替换 NMS。DIoU_NMS 认为若相邻矩形框的中心点更靠近当前最大分数矩形框的中点，这些长方形框很可能是多余的，它不但考察了重合的范围，还考察了两个矩形框中心点间的距离。DIoU_NMS 公式如

下：

$$s_i = \begin{cases} s_i, & DIoU(M, b_i) < \varepsilon \\ 0, & DIoU(M, b_i) \geq \varepsilon \end{cases} \quad (3-13)$$

DIoU_NMS 引入参数 β ，当 $\beta \rightarrow \infty$ 时，DIoU 与 IoU 效果相当。当 $\beta \rightarrow 0$ 时，保留不与中心点重合的框。DIoU 的定义如下：

$$DIoU = IoU - \left(\frac{d^2}{c^2}\right)^\beta \quad (3-14)$$

式中： d 为两个矩形框中心点距， c 为两个矩形框之间的最长对角线长度。

DIoU_NMS 考虑中心点有助于缓解遮掩与重叠的案例，而且在保持 NMS 阈值不变的情况下，肯定可以保留更多的矩形框，能够更好的预测目标物体。同时 DIoU_NMS 还可以与其他 NMS 变体使用，如加权平均法、定位优先法和得分惩罚法等。本文使用的 DIoU_NMS+CIoU_Loss 结合使用，替换后的 YOLOv5 算法在精度有很大的提升。

4) 多尺度检测的改进

从图 3-8 可看出，YOLOv5 使用了 3 个输出检测头，分别负责预测大中小目标。由于本文使用的数据集 TT100K 中，大多交通标识只占整张图片的 1%~2%，这就导致这些交通标识在大目标检测头和中目标检测头中只有几个像素点，丢失大量位置信息，甚至会在多次卷积中全部丢失。因此，在 Neck 模块中增加一个输出特征图大小为 160×160 的目标检测头，提高对小目标的检测精度，改进后的网络结构如图 3-11 所示。

Neck 模块将 80×80 大小的特征图的 C3 模块进行上采样操作后，与 Backbone 模块的第 2 层特征图进行 concat 融合，然后通过 Conv 层操作后向两个模块输出，一个是作为一个新的检测头，用来检测 4×4 大小的交通标识，另一个是进行下采样，进行加权特征融合。从图 3-13 中可以看出，大小为 160×160 的目标检测头在整个结构的最后，更能提取到完善的语义信息，因此为输出层增加位置信息尤为重要。增加小目标检测头后的 YOLOv5 算法在精度上有着显著提升。

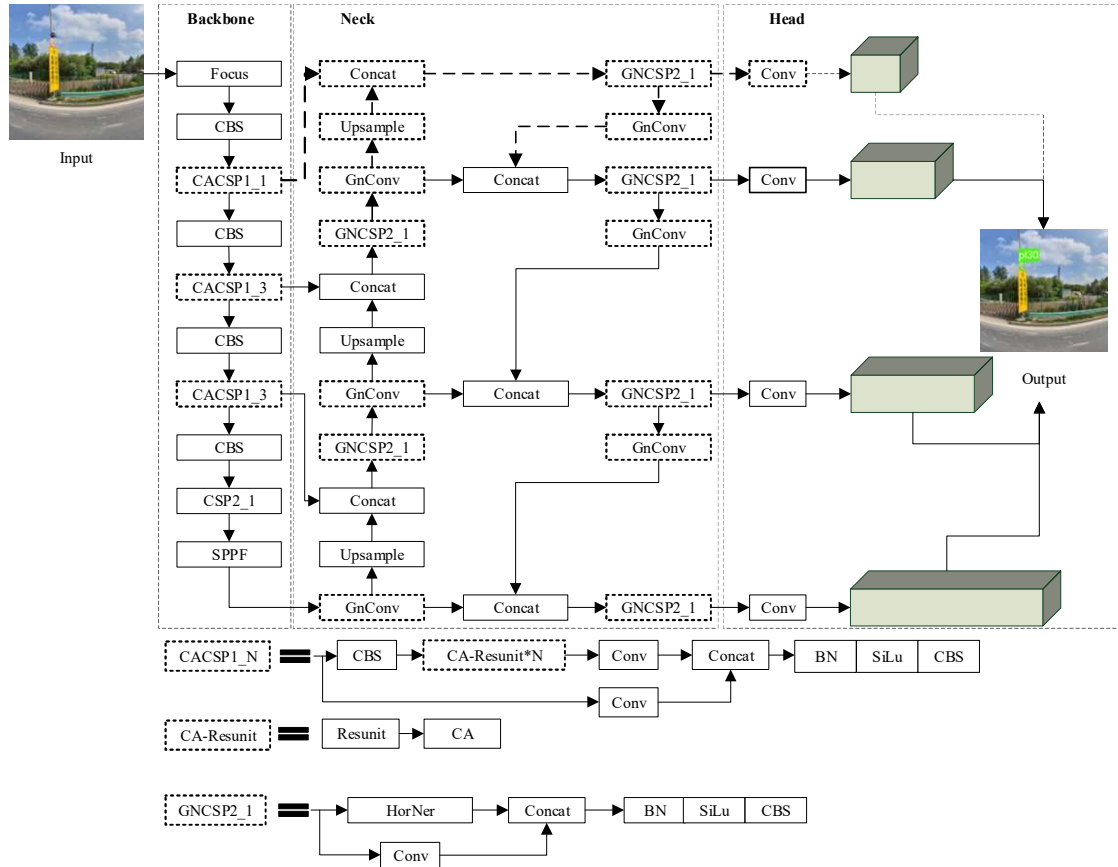


图 3-11 改进后的 YOLOv5 网络结构

3.3 模型实验验证及结果分析

3.3.1 实验参数及准备

1) 数据集准备

本文使用的数据集是腾讯和清华联合制作的 TT100K 交通标志数据集，包含了 10 万张图片，221 类交通标志。但是，里面只有 1 万多张图片包含交通标志，经过统计，训练集和测试集中实例分别为 16527 和 8190。超过 100 张的实例只有 45 个类别，因此本文选取这 45 类交通标志进行训练，图 3-12 为 45 个类别的实例分布。

首先为了让数据集能够在 YOLOv5 中进行训练，将 TT100K 数据集转化成 VOC 格式，并且用 Python 脚本选出 45 类超过 100 张的图片和 XML：首先创建标准的 VOC 文件夹，然后用 Python 脚本生成整个数据集的 XML 文件并删除 txt 中 id 对应的 XML 图片，最后生成 train.txt 和 val.txt 文件，转换成功后再将 VOC

格式转化为 YOLO 格式的数据集。经过处理的图像集共有 9170 张，训练集和测试集的比例为 8:2。

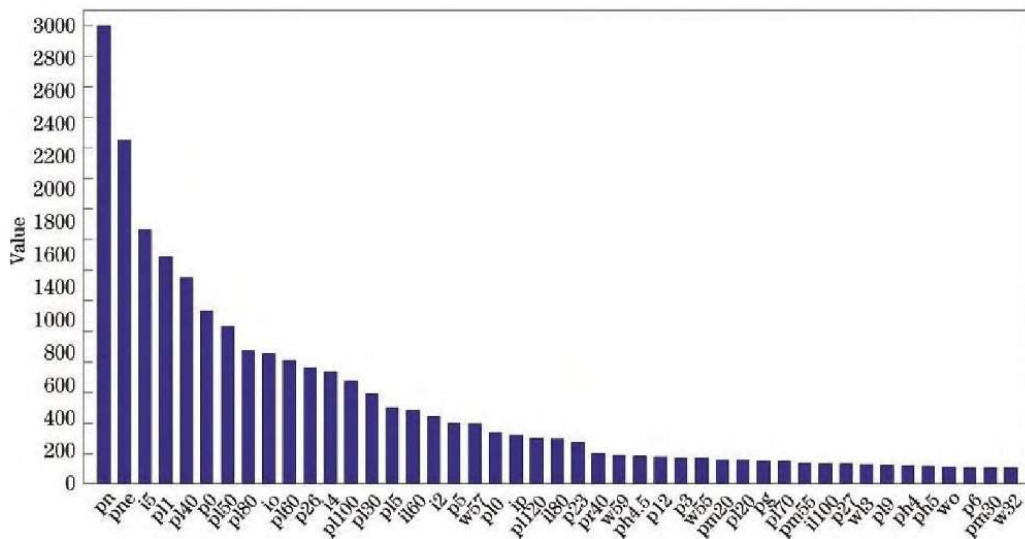


图 3-12 TT100K 中超过 100 张的类别

2) 实验环境及参数设置

由于使用的数据集较大，因此在 GPU 上对模型进行训练，环境配置见表 3-1：本文在 YOLOv5s 模型上进行改进，参数配置为：批量大小为 128，线程数为 8，训练轮数为 300，初始学习率为 0.01。

表 3-1 实验环境配置表

配置名称	版本及型号
操作平台	Ubuntu18.04
GPU	NVIDIA RTX A5000 (24G) ×2
CPU	INTEL Xeon 5218R×2
内存	128G
深度学习框架	Pytorch1.12.1
CUDA&Cudnn	v11.3&v8.3.2

3) 评价指标

本文基于准确率 P、召回率 R 和平均精度均值 mAP 评估原始的 YOLOv5s 模型和改进后的模型。改进后的 YOLOv5 算法各指标曲线如图 3-13 所示。

P 是指预测为正的正样本比上所有预测为正的样本，R 是指预测为正的样本比上所有正样本，平均精度均值 mAP 是平均精度 AP 的均值，计算公式如下：

$$P = \frac{TP}{TP + FP} \quad (3-15)$$

$$R = \frac{TP}{TP + FN} \quad (3-16)$$

$$AP = \int_0^1 P(R) dR \quad (3-17)$$

$$mAP = \sum_{i=1}^N \frac{AP_i}{N} \quad (3-18)$$

式中： TP 是预测为正的正样本， FP 是预测为正的负样本， FN 是预测为负的正样本， N 表示总类别数。

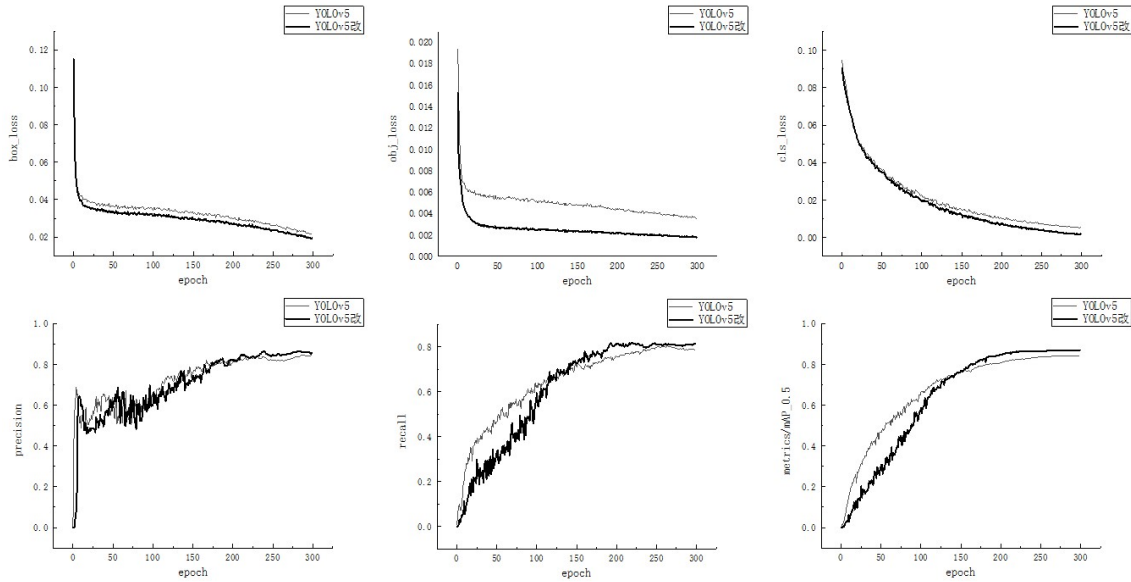


图 3-13 改进后模型的各项指标曲线

3.3.2 实验结果验证与分析

为了验证本文算法中各模块的作用，分析各模块对算法精度影响，本文进行了消融实验。以 YOLOv5-7.0 为基础网络逐步增加改进点，实验结果如表 3-2 所示。根据表中数据显示，在 Backbone 中的 C3 中加入了 CA 注意力机制后，可以更加精准地定位和识别感兴趣的目标，mAP 提升了 0.43%。在将 DIoU_NMS 引入后，有助于缓解遮挡和重叠案例，mAP 提升了 0.37%。加上小目标检测头后，更能提取到完善的语义信息，mAP 提升了 3.03%。本文在同时引入 CAC3、DIoU_NMS 和小目标检测头，P 值提升了 1.79%，R 值提升了 2.52%，mAP 值提升了 2.79%，达到了 86.96%，验证了本文算法的有效性。

表 3-2 YOLOv5 和改进的 YOLOv5 消融对比

YOLOv5s	CAC3	DIOW_NMS	HorNet	小目标 检测	P(%)	R(%)	mAP_0.5(%)
√					83.67	79.00	84.17
√	√				84.25	79.08	84.60
√		√			83.81	80.31	84.54
√			√		84.21	80.52	85.38
√				√	87.44	80.39	87.20
√	√	√			84.99	79.54	84.87
√	√	√	√		85.05	80.23	85.02
√	√	√	√	√	85.46	81.52	86.96

为了进一步体现出算法的精确性，从 TT100K 中选出 3 张图片进行测试，对比原始 YOLOv5 算法与改进后的 YOLOv5 算法，如图 3-14 所示。第一幅图像是复杂的街道场景，原始的 YOLOv5 算法有漏检现象，经过改进后基本都检测到了。第二张图片的背景较为简单，但目标都较小，原始 YOLOv5 算法不但漏检许多，还出现了错检，而改进后的算法依旧保持着优良的精度。第三张图片中交通标志遮掩较多，原始 YOLOv5 算法没有检测出重叠后的类别 w32，可以看到本文算法对于小目标的检测性能适用于大多复杂的背景。



图 3-14 改进 YOLOv5 算法检测结果对比

本文将提升后的 YOLOv5s 模型对比了 Faster-RCNN、SSD、YOLOv3 和 YOLOv4 结果，如表 3-3 所示。从表中可看出，本文算法与 Faster R-CNN 相比，mAP 提升了 12.38%，与单阶段算法 SSD、YOLOv3、YOLOv4 算法相比，mAP 分别提升了 16.5%、13.32%、13.05%。综上所述，与其他主流算法相比，本文算法在交通标志检测精度方面具有显著优势。

表 3-3 不同算法检测结果对比

算法	P(%)	R(%)	mAP_0.5(%)	mAP 提升(%)
Faster R-CNN	75.54	75.20	74.58	12.38
SSD	69.56	70.28	70.46	16.5
YOLOv3	77.96	66.07	73.64	13.32
YOLOv4	81.12	66.82	73.91	13.05
YOLOv5 (改)	85.46	81.52	86.96	

3.4 本章小结

本章概述了深度学习中重要的卷积神经网络结构和 Darknet 深度学习框架，并提出了改进的 YOLOv5 算法。针对数据集中变化尺度的问题，本文在骨干网络的 C3 中加入 CA 注意力机制，增强算法对于目标的精确定位能力；针对传统 NMS 在识别重叠区域时出现遗漏和错误的问题，引入 DIoU_NMS 来保证重叠遮掩部分的精度；加入了 gⁿConv 执行高阶空间交互；针对原始 YOLOv5 算法对于小目标检测不出的问题，在原来三输出头基础上，增加一个小目标检测头，提高小目标的识别精度。改进后的算法在公共数据集上测试，对比原始算法精度明显提升，并且数据量很小，适合运用到实践中。

第 4 章 移动机器人导航算法设计

自主导航的关键在于为移动机器人设计一条平滑安全的路径，保障其安全地到达指定目的地。本章节将介绍采用改进的 PRM 算法进行全局路径规划，同时使用结合了深度学习和 TEB 算法的方法进行局部路径规划，通过一系列的实验验证确立了这两种路径规划方法的有效性。

4.1 基于改进 PRM 的全局路径规划算法设计

4.1.1 PRM 算法基本原理

PRM 算法是一种移动机器人路径规划算法，主要用于解决二维度配置空间的路径规划问题，算法分为两个阶段：学习阶段和查询阶段。在学习阶段，PRM 随机采样生成路标节点，检查它们是否位于障碍物之外，并将这些节点连接起来形成一张图。查询阶段时，会在图中搜索从起点到目标点的路径，并采用 A* 算法来寻找最优路径。PRM 算法适用于复杂环境中的多次查询，因其易于实现且能够有效处理复杂场景而受到青睐。

传统 PRM 算法如图 4-1 所示，图中 q_{start} 为起始点， q_{goal} 为目标点，蓝线为构建的路径网络图，红线为最终规划的路径。

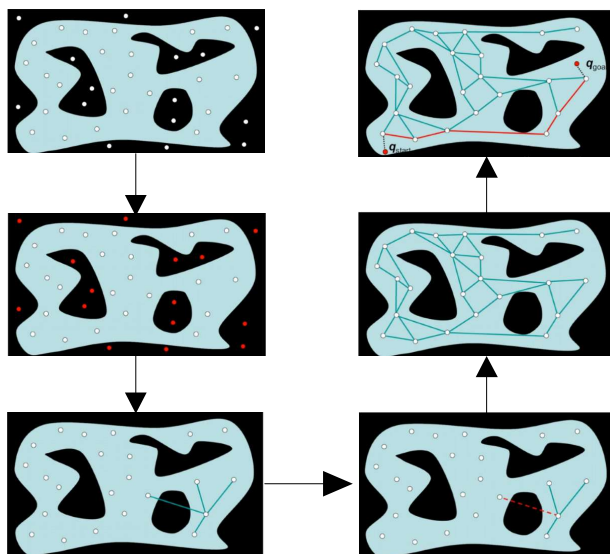


图 4-1 传统 PRM 算法

传统 PRM 算法在地图中存在狭窄通道时，均匀采样方法的效果就会大打折扣。由于采样密度在全图范围内均等，而采样点的数量与区域体积成正比，导致狭窄区域的采样点不足，难以在路径网络图中形成有效连线，以至于无法规划出最优路径。

4.1.2 PRM 算法的改进

传统 PRM 算法在整个图内均匀地分布采样点，并将那些位于障碍物上的点直接排除，这可能导致需要重复采样且降低算法效率。为了改善这一局限性，引入人工势场法优化采样策略，利用人工势场法可以使采样点在势场中运动这一思想，把构建的地图看成一个虚拟力场，虚拟力场是由自由空间对采样点的引力场和障碍物内部对采样点的斥力场组成，这样能够将落在障碍物内的采样点移动到自由空间，从而提高采样效率并减少无效采样的次数。

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (4-1)$$

其中： q 为采样点， $U(x)$ 为势场和， $U_{att}(x)$ 为引力场， $U_{rep}(x)$ 为斥力场

采样点在虚拟势场中的动能公式如下：

$$\begin{aligned} E_q &= \int_D F(r) dr \\ D &= x(t)i + y(t)j \end{aligned} \quad (4-2)$$

式中： $F(r)$ 为采样点受到的合力， D 为采样点的轨迹， $x(t)y(t)$ 为 D 在 ij 方向的分量。

在已知地图的势场分布以及初始采样点位置的情况下，可以计算出采样点在势场影响下最终的静止位置。这种方法使得最初位于障碍物中的采样点被重新定位到自由空间内，这不仅维持了所需的采样点数量，还特别提升了狭窄区域的采样密集度，这种策略有效地增强了算法的工作效率。整体算法流程图如图 4-2 所示。

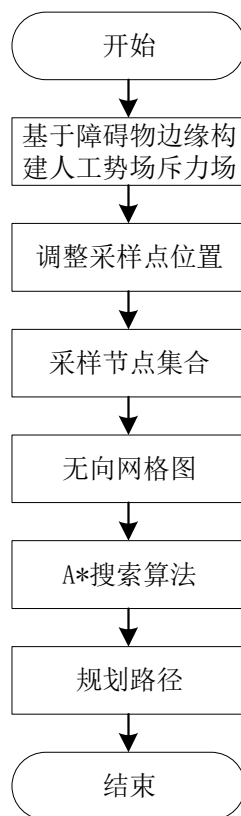


图 4-2 改进 PRM 算法流程图

4.1.3 改进的 PRM 算法测试结果与分析

为了检验改进算法在狭窄通道上采样的有效性,本文在 Matlab2023a 环境下对其采样阶段和整体路径规划效率进行了仿真实验。首先建立一个含有障碍物的 $500*500$ 的地图,为便于仿真,将无人车视为一个质点。改进前后的 PRM 算法仿真如图 4-3 所示,两种算法试验结果对比见表 4-1。

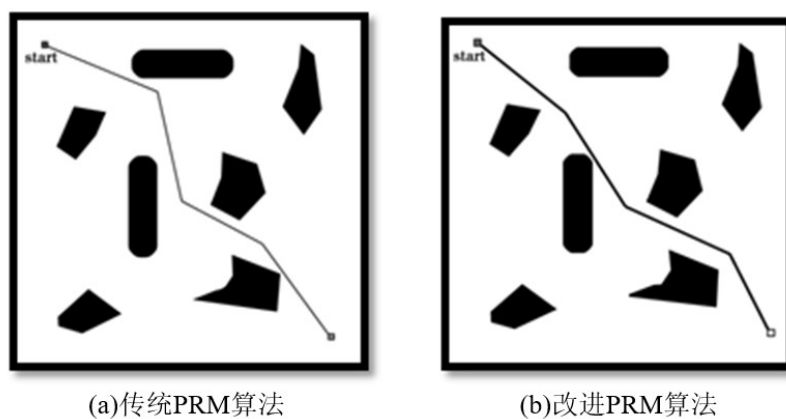


图 4-3 改进前后 PRM 算法对比

表 4-1 改进前后算法对比

算法	转折点数	路径长度	平均运行时间/s
传统 PRM	3	705	0.66
改进 PRM	1	480	0.29

由实验结果可知，改进后的 PRM 算法仿真试验所得到的路径轨迹转折点较少，路径轨迹更加平滑。路径长度的缩短减少了路径规划所需的时间，移动机器人能顺利到达目标位置，提高了搜索效率，实现了路径优化目标

4.2 局部路径规划算法设计

4.2.1 TEB 算法基本原理

EB 算法^[54]是一种用于移动机器人路径规划的方法，其核心思想是创建一个动态调整的路径或“弹性带”。该算法从初始路径开始，随后根据障碍物的存在和移动机器人的实时位置，通过吸引力和排斥力相结合的方式，调整并优化路径。此过程类似于弹性带在障碍物周围弯曲和伸缩，保证了移动机器人路径的平滑性和安全性。

EB 算法由 n 个的移动机器人位姿 $s_i = [x_i, y_i, \beta_i]^T$ 组成的序列，其中 x_i, y_i 是移动机器人位置， β_i 为全局中移动机器人的朝向。序列如式（4-3）所示

$$Q = \{s_i\}_{i=0 \dots n} \quad (4-3)$$

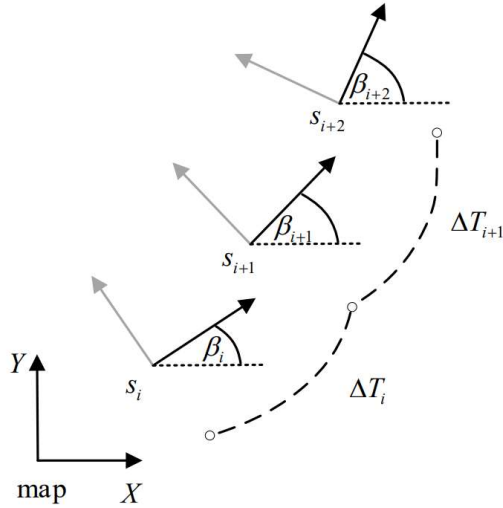
TEB 算法^[55]在经典 EB 算法上引入了时间维度，并且将移动机器人的路径可行性问题纳入了考虑范围，针对移动机器人所受到的各种动态约束，用最少的时间来找最快的避障路径。它设定好时间差 δT_i 来表示相邻两个路径点的行进时间，公式如下：

$$\tau = \{\delta T_i\}_{i=0 \dots n-1} \quad (4-4)$$

则算法的优化序列公式如下：

$$B := (Q, \tau) \quad (4-5)$$

TEB 结构图如图 4-4 所示。



4-4 TEB 结构图

TEB 算法的主要思想是把姿态和时间间隔作为优化目标，通过施加一系列动态约束来计算最优局部路径轨迹。然后通过对这些约束进行加权求和，得到总体约束函数 f_b 。 f_b 公式如下：

$$f_b = \sum_0^n \frac{1}{[d(x_i, o_m)]}, m = 0, 1, \dots, n; \quad (4-6)$$

通过对两个时间间隔 ΔT 的连续位姿及其间距和角度的改变，得到了移动机器人的线速度和角速度。线速度，线加速度公式如下：

$$v_i \approx \frac{\begin{pmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{pmatrix}}{\Delta T_i} \quad (4-7)$$

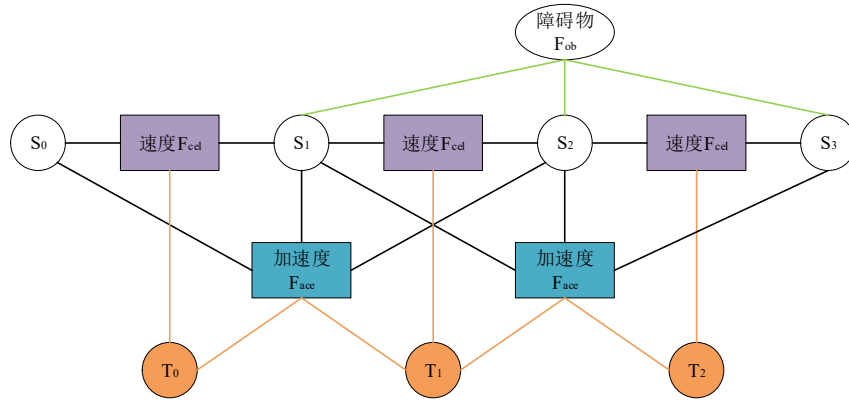
$$a_i \approx \frac{2(v_{i+1} - v_i)}{\Delta T_i + \Delta T_{i+1}} \quad (4-8)$$

角速度，角加速度公式如下：

$$\omega_i \approx \frac{\beta_{i+1} - \beta_i}{\Delta T_i} \quad (4-9)$$

$$a_i \approx \frac{2(\omega_{i+1} - \omega_i)}{\Delta T_i + \Delta T_{i+1}} \quad (4-10)$$

TEB 算法处理稀疏系统问题时，通过以位姿和时间间隔作为节点的超图表示。优化目标和约束的代价函数在这个超图中表示为节点之间的边，通过这些边连接各个节点，形成一个超图结构。如图 4-5 所示。



4-5 TEB 算法超图

在超图中，速度约束连接机器人连续位姿 S_i 和 S_{i+1} 和时间间隔 T_i ；加速度约束连接机器人连续位姿 S_{i-1} 、 S_i 和 S_{i+1} 和连续时间间隔 T_i 和 T_{i+1} ；障碍物约束 F_{ob} 连接障碍物和距离障碍物最近的机器人位姿 S_i 。

4.2.2 TEB 路径规划算法仿真

为验证移动机器人应用 TEB 进行局部路径规划的可行性，进行实验仿真。如表 4-2 所示为仿真实验参数

表 4-2 仿真实验重要参数

参数项	数值	参数项	数值
障碍物膨胀半径	0.5m	障碍物最小安全距离	0.25m
线速度	0.5m/s	角速度	0.3rad/s
线加速度	0.5m/s ²	角加速度	0.5rad/s ²

通过 TEB 算法获得的移动机器人行进轨迹，并观察了面对不同数量的障碍物时，TEB 算法优化后的仿真效果，如图 4-6 所示。

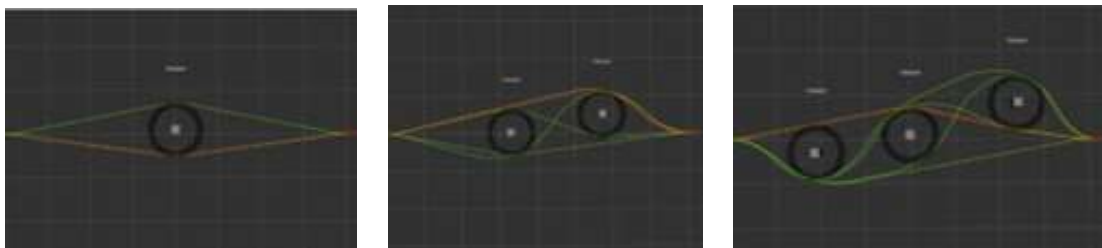


图 4-6 TEB 仿真结果

针对每个障碍物数量 n ，TEB 算法可以生成 $2n$ 条局部路径，并根据设定的权值和控制参数，选择出最优的路径。在仿真中，障碍物用灰色框表示，膨胀

半径为黑色部分，最优路径用红线表示，由图可见 TEB 算法能够在众多备选路径中选出最优路径来绕过障碍物。

4.2.3 TEB 算法结合深度学习避障设计

1) 整体避障策略

本研究结合了深度学习技术和 TEB 局部路径规划算法来实现移动机器人的局部避障，该方法能够使移动机器人准确地识别障碍物并进行避障。首先使用深度学习来识别障碍物并确定其位置，移动机器人根据识别结果采取相应行动：如障碍物位于左侧，移动机器人将往右避让；若障碍物位于右侧，移动机器人则往左避让。当感知到障碍物过大时，会被判断为过于靠近障碍物，此时移动机器人选择后退以保持安全距离。

局部路径规划流程如图 4-7 所示。深度学习识别的障碍物类别与位置信息传输给 TEB 算法进行局部路径规划，局部路径规划器通过 TEB 算法规划的避障路径来执行局部避障操作。

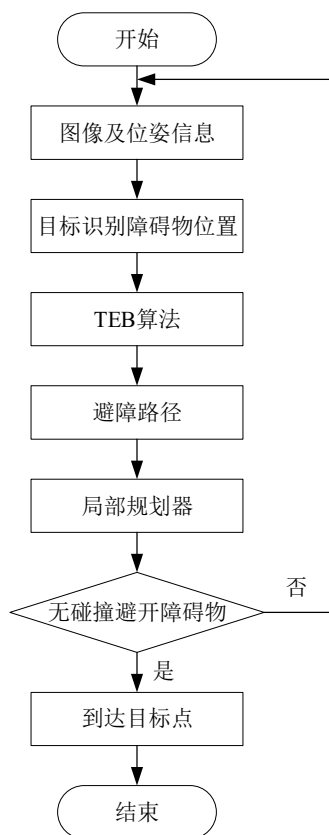
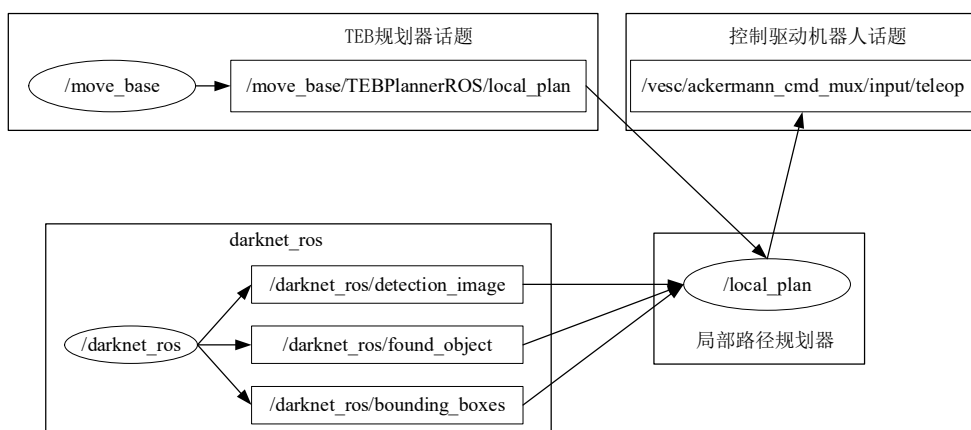


图 4-7 局部路径避障流程图

2) 避障策略的实现

整体避障策略的思想是深度学习识别出障碍物后，对识别框的位置进行分析，结合 TEB 算法制定避障策略，生成详细的避障轨迹。由于移动机器人搭载了 ROS 框架，且 TEB 算法的开源功能包可以直接调用，所以通过 ROS 的通信机制集成实现。ROS 环境各节点间的关系如图 4-8 所示。



4-8 避障策略 ROS 节点关系图

Darknet_ROS 框架通过订阅 `/imx/rgb/image\_raw\_color` 来获取图像，对障碍物进行识别。识别出的障碍物位置信息实时发布至可以估算障碍物位置信息的 `/bounding\_boxes` 话题。`/local\_plan` 融合识别的位置信息与 TEB 算法的规划结果，发布最终的控制指令到驱动移动机器人话题，通过该话题控制移动机器人执行移动操作，以实现避障。

4.3 融合全局与局部路径规划算法

4.3.1 混合路径规划

在现实环境中经常遇到未知障碍物的情况，如果这些障碍物位于全局路径上，会导致移动机器人与其碰撞。因此在实际应用中，采用混合路径规划方法，这种方法结合了全局路径规划和局部路径规划，以确保移动机器人能够有效避开障碍物并安全导航。混合路径规划导航如图 4-9 所示。

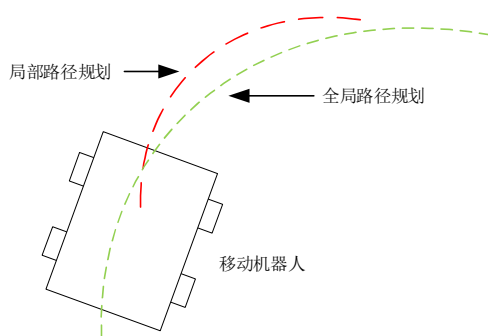


图 4-9 混合路径规划

当移动机器人在实际环境中运行时，通过改进的 PRM 算法来进行全局路径的规划，同时移动机器人会采集周边环境信息，如果在全局路径上检测到障碍物，使用深度学习进行目标识别和 TEB 算法结合来规划避障路径。一旦安全绕过障碍物并到达安全区域，移动机器人再次将全局路径设为目标进行移动。通过这一连续的步骤，移动机器人能够安全且高效地抵达目的地。流程如图 4-10 所示：

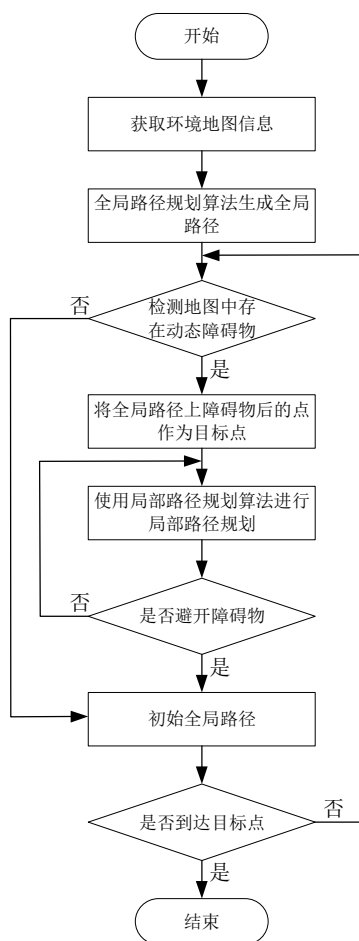


图 4-10 混合路径规划流程图

4.3.2 混合路径规划仿真实验

如图 4-11 (a) 所示, 添加未知障碍物到仿真环境中, 图 4-11 (b) 为移动机器人在模拟室内环境下采用混合路径规划避开障碍物的结果, 图中红色箭头所指位置为新增障碍物, 由激光雷达扫描得到。

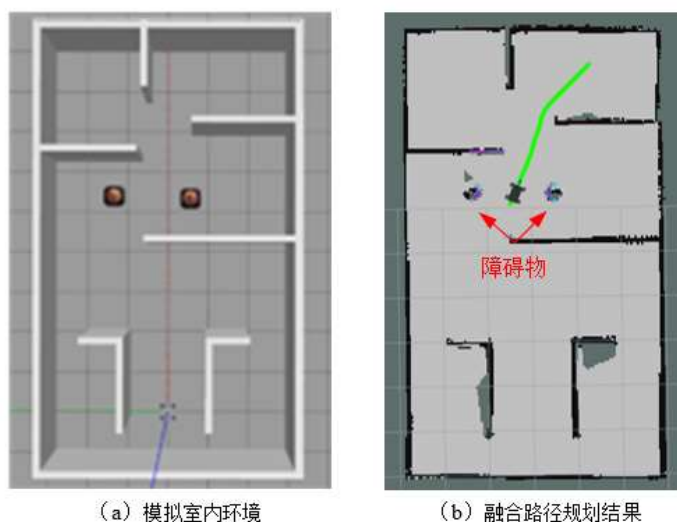


图 4-11 仿真实验

上图中路径展现了移动机器人在通过激光雷达探测到障碍物时能够及时做出调整并绕过障碍物, 即使在障碍物的数目、位置变动以及环境地图规模改变的情况下, 混合路径规划策略依旧能有效规划出安全的路径。因此, 通过仿真实验可以看出, 混合路径规划不仅具备了避障能力, 而且也能实现全局最佳路径的规划。

4.4 本章小结

本章对传统 PRM 算法在狭窄空间规划路径时的不足进行了改进, 提出了一种基于优化采样策略的 PRM 算法, 有效增加了狭窄区域的采样点数, 增强了该算法在狭窄区域中规划路径的能力。此外, 结合 TEB 算法和深度学习应用到局部路径规划, 在深度学习标识障碍物后进行初步避障判断, 紧接着 TEB 算法根据这些定位进行精确的路径调整。为了适应复杂环境, 将全局和局部路径规划结合形成混合路径规划策略。最后通过构建仿真环境, 验证了本文所提出路径规划算法的有效性。

第 5 章 实验与分析

本章将对移动机器人自主导航系统进行整合，在搭建的移动机器人上进行实验分析。首先本章介绍了移动机器人硬件和软件部分设计，然后进行深度学习模型的部署，最后依据构建的地图，进行移动机器人的路径规划实验和自主避障实验。

5.1 实验平台整体设计

5.1.1 硬件平台设计

本文移动机器人实验平台使用航天科工集团第三研究院设计的轻舟移动机器人，车身配备里程计、IMU、激光雷达等传感器，并使用 JETSON NANO 主板进行控制和数据处理，控制器的外部接口可用于连接传感器及 PC 端。如图 5-1 所示。



图 5-1 移动机器人实验平台

轻舟移动机器人的平台结构图如图 5-2 所示。

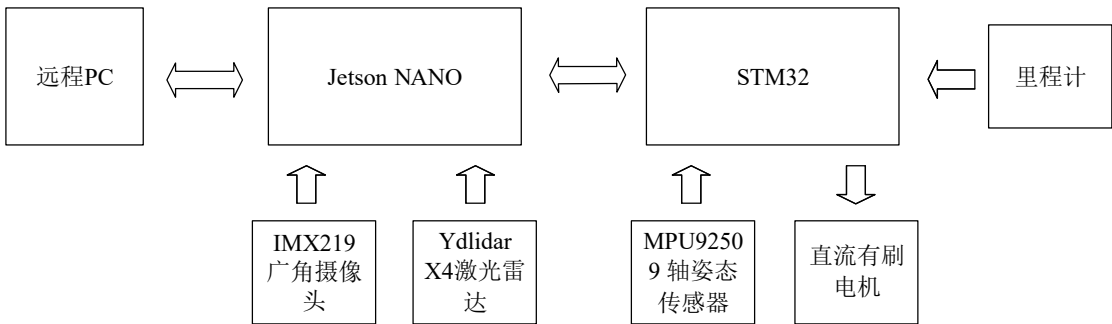


图 5-2 移动机器人平台结构图

车载主机采用了 NVIDIA Jetson NANO 主板，如图 5-3 所示。其预装 Ubuntu 18.04 LTS 系统，搭载英伟达研发的 128 核 Maxwell GPU。

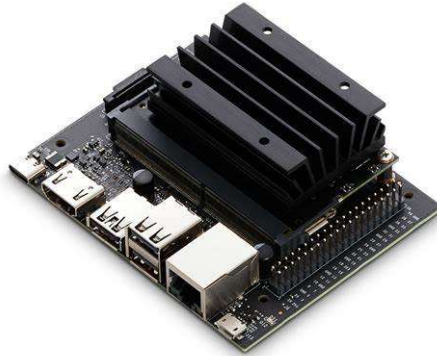


图 5-3 Jetson NANO

Jetson NANO 参数如表 5-1 所示：

表 5-1 Jetson NANO 参数

配置名称	数值
GPU	NVIDIA Maxwell w/128 NVIDIA CUDA 核心
CPU	四核 ARM Cortex-A57 MPCore 处理器
内存	4GB 64 位 LPDDR4
显卡	HDMI 和 DisplayPort 输出
通讯方式	USB、WiFi、蓝牙

远程 PC 为主机电脑，主要用来远程控制移动机器人。STM32 控制主板主要完成轻舟移动机器人前轮转向舵机和后轮驱动电机的运动控制，同时支持多种功能。车载主机和远程 PC 搭载了 Ubuntu 18.04 操作系统，ROS 版本为 Melodic Morenia。

移动机器人平台搭载了 10 米测距、12Hz 扫描频率的 Ydlidar X4 激光雷达，800 万像素、160 度广角摄像头的 IMX219 单目摄像头，采用直流有刷电机，通过舵机和连杆悬挂实现阿克曼转向，搭载 MPU9250 9 轴姿态传感器（3 加速度、3 陀螺仪、3 磁力计）和增量脉冲计数的里程计获取移动机器人实时的位姿信息。

如图 5-4 所示，在实验室里划出场地自主搭建环境，使用挡板隔开组成路径作为实验的场景。



图 5-4 实验环境

5.1.2 软件功能模块设计

在本文中，所有实验均通过运行在 Ubuntu 18.04 操作系统下的 ROS 框架来实施。实验平台包括轻舟移动机器人和配置了相同操作系统的远程控制计算机，二者连接在同一局域网内。依靠 SSH（安全壳层协议）远程访问机制，研究人员可以从远程控制计算机上登入并操作移动机器人的系统。整体分为以下几层：

- （1）应用层：位于架构顶层，主导整个软件算法的设计工作。
- （2）传输层：负责数据的高效传输，包含数据的封装和解包过程。
- （3）驱动层：负责实现硬件的驱动功能，其主要负责驱动移动机器人所需的传感器设备，保证硬件设备的正常运行和数据的实时采集。。
- （4）硬件层：作为系统中实装的物理设备基底，包含有传感器设备和主板等元件。

四层架构降低了系统组件间的耦合度，增加了软件的整体稳定性，并提升了移动机器人运行的稳定性。本文移动机器人 ROS 节点关系如图 5-5 所示。

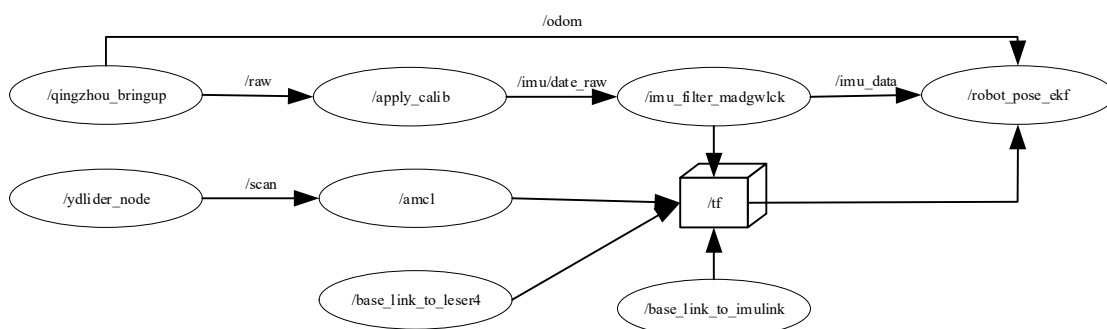


图 5-5 底层驱动运行节点关系图

5.2 移动机器人深度学习模型部署

5.2.1 模型部署

Darknet_ROS 订阅摄像头话题来接收实时图像数据，随后利用深度学习模型对这些图像进行目标检测，将检测到的目标信息即时发布。移动机器人通过订阅这一话题接收到障碍物的信息，并通过 TEB 算法分析确定障碍物的精确位置，从而进行相应的避障动作。ROS 节点运行关系如图 5-6 所示。

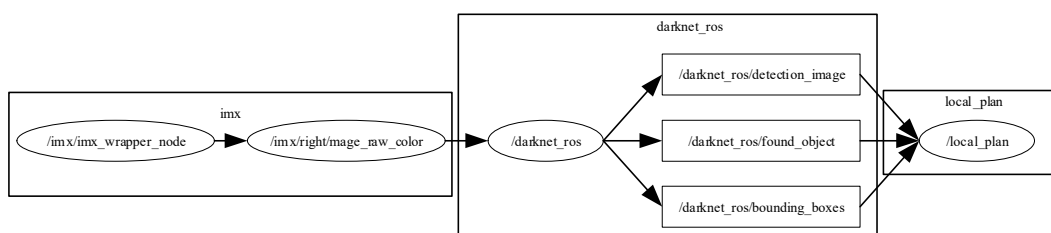


图 5-6 ROS 节点运行关系图

Darknet_ROS 通过订阅 `/imx/right/image\_raw\_color` 话题接收实时图像，然后输入图像到深度学习模型进行障碍物识别和检测，并将结果发布到三个话题：`/found\_object`、`/bounding\_boxes` 和 `/detection\_image`。`/found\_object` 话题记录对象数量，`/bounding\_boxes` 话题记录障碍物的位置尺寸，`/detection\_image` 话题发布有检测框的图像。移动机器人订阅这些话题获取处理后的信息，并根据这些数据采用 TEB 算法执行局部避障。

5.2.2 YOLOv5 实时测试

本文通过自制数据集对改进模型进行训练，数据集里的图片是通过将摄像头采集的视频流进行分帧操作得到的，并对目标障碍物进行标注分为三个类别。标注障碍物如图 5-7 所示。



图 5-7 障碍物标注

将改进的深度模型导入到 Darkner_ROS 框架，测试移动机器人实时目标识别的效果，识别结果如图 5-8 所示。



图 5-8 实时测试结果

5.3 路径规划实验

在实验室搭建的地图上进行全局和局部路径规划，在构建完成的环境地图中进行全局路径规划实验，图 5-9 为改进前后的全局路径规划结果的对比图：

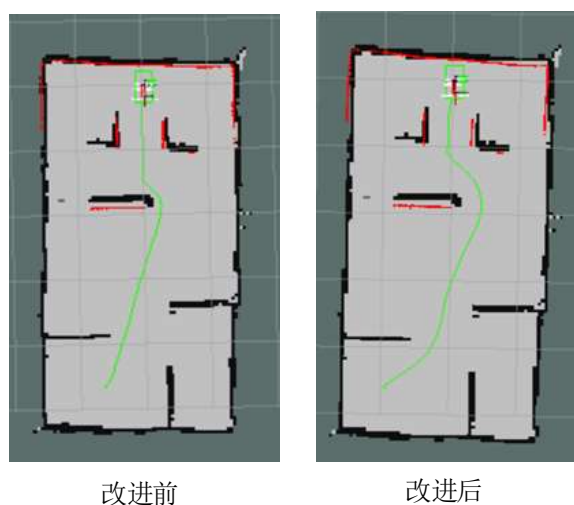


图 5-9 全局路径规划效果

为对环境地图的路径规划进行对比，提取图 5-9 中改进前后的全局路径，获得图 5-10 的路径对比结果。从图中得出，本文设计的全局路径得到较平滑的路径。

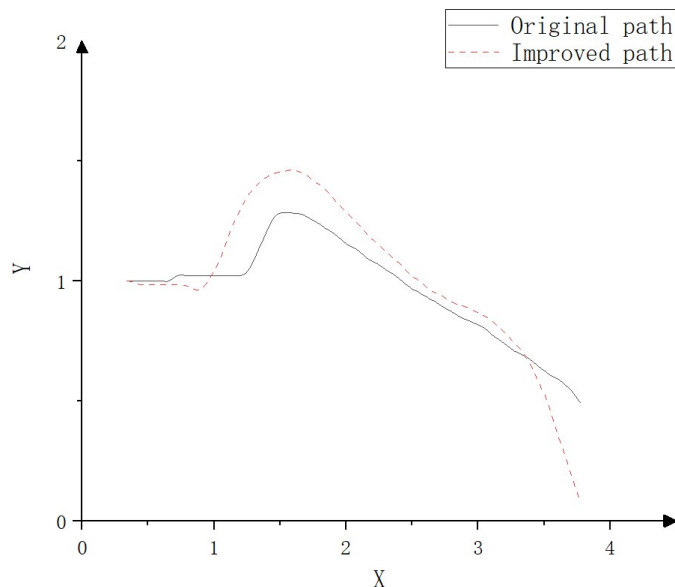
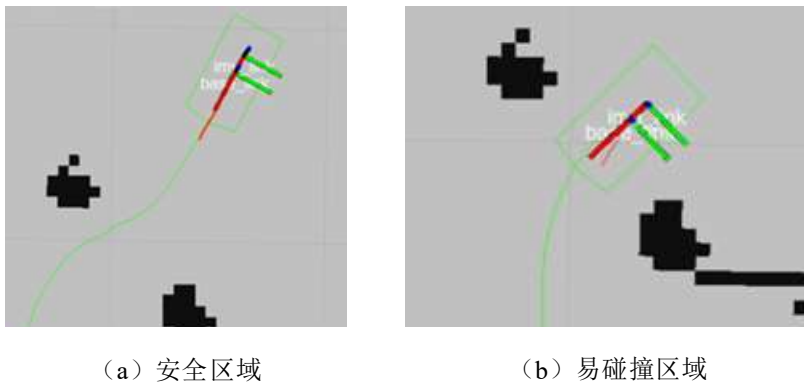


图 5-10 全局路径规划改进前后对比

在构建完成的环境地图中进行深度学习结合 TEB 局部路径规划，局部路径规划如图 5-11 所示。在图 5-11 (a) 中移动机器人所处区域较为安全，故局部路径规划遵循全局路径规划，在图 5-11 (b) 中移动机器人所处易碰撞区域，为保证移动机器人的安全性，将减小贴近全局路径的权重。



(a) 安全区域

(b) 易碰撞区域

图 5-11 局部路径规划

5.4 自主避障实验

将针对自制数据集训练完成的深度学习网络模型部署到移动机器人平台，

并在室内实验室进行实验。通过手动控制移动机器人进行建图，设置路障 B 为临时障碍物，该障碍物在地图创建时并不存在。在有无临时静态障碍物情况两种实验场景下分别进行导航实验，移动机器人移动过程中，可利用 `rviz` 工具实时监控移动机器人的位姿以及路径规划的情况。建图结果如图 5-12 所示。

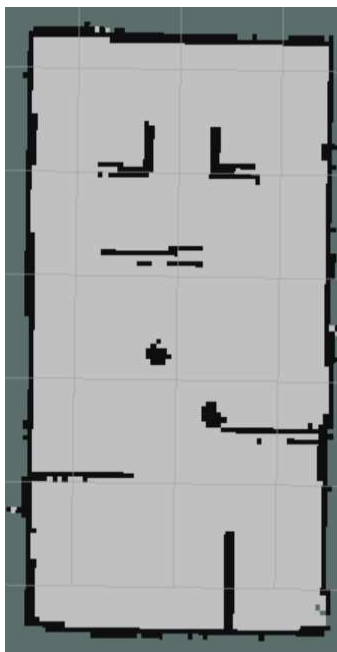


图 5-12 激光 SLAM 建图

启动移动机器人上的 ROS 并运行 `Darknet_ROS` 包之后，使用 `rviz` 设置移动机器人的起始位置和目标点。在移动机器人移动时，可以观察它是否能准确识别出不同类别的障碍物，并在发现障碍物后执行避障操作，以顺利到达预定目标位置。

1) 无静态障碍物

将构建的地图作为移动机器人的全局地图，设置起点和终点对移动机器人进行验证。移动机器人位姿如图 5-13 所示，图 5-13 (a) 准备经过障碍物区，图 5-13 (b) 根据全局路径规划进行转弯，图 5-13 (c) 运行到两固定障碍物中间，图 5-13 (d) 到达预定目标位置。

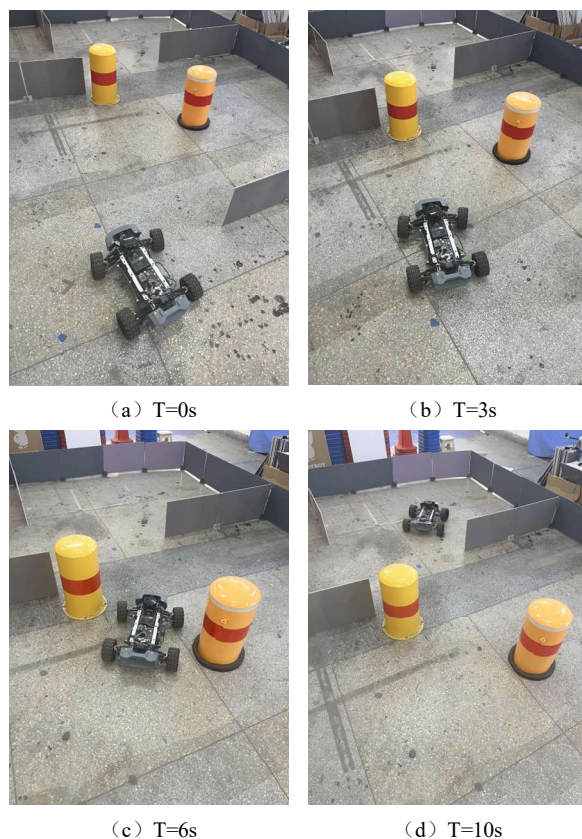


图 5-13 无静态障碍物实验过程

移动机器人运行过程中识别障碍物如图 5-14 所示，图 5-14 (a) 为移动机器人在开始经过障碍物区的初始位置识别 A 障碍物在右方，但是距离过远依然按照全局路径规划的线路行进，图 5-14 (b) 为移动机器人识别出 C 障碍物，但是距离过远依然按照全局路径规划的线路行进，图 5-14 (c) 为移动机器人即将接近 A、C 障碍物，移动机器人准备进行转弯避障。

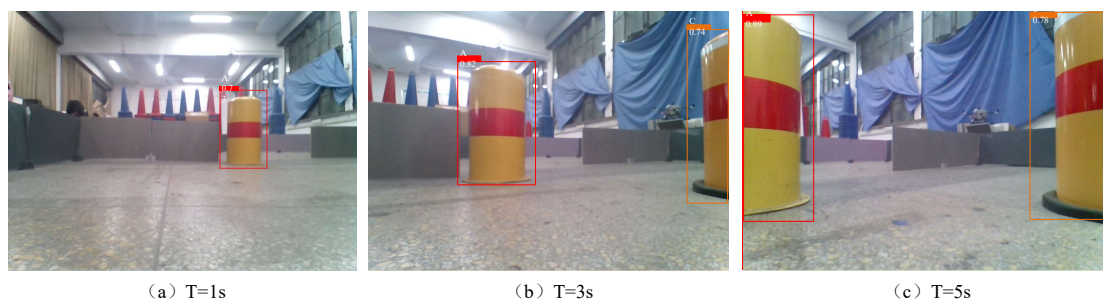


图 5-14 障碍物识别过程

在 rviz 中，移动机器人运行的路径轨迹如图 5-15 所示。



图 5-15 移动机器人轨迹

2) 存在静态障碍物

当场景中存在静态障碍物时，基于深度学习的目标识别能够判定障碍物大致位置，从而为移动机器人做出预避障指令。接着通过 TEB 算法判断避障指令是否正确，确保在行进过程中移动机器人能够避免发生碰撞，最后将深度学习与 TEB 算法结合完成整个避障流程。移动机器人位姿如图 5-16 所示，其中图 5-16 (a) 准备经过障碍物区，图 5-16 (b) 根据全局路径规划进行转弯，图 5-16 (c) 判断无法通过停止行进，图 5-16 (d) 退回到上一次转弯的位置，图 5-16 (e) 选择新的路线进行避障，图 5-16 (f) 绕过障碍物区。

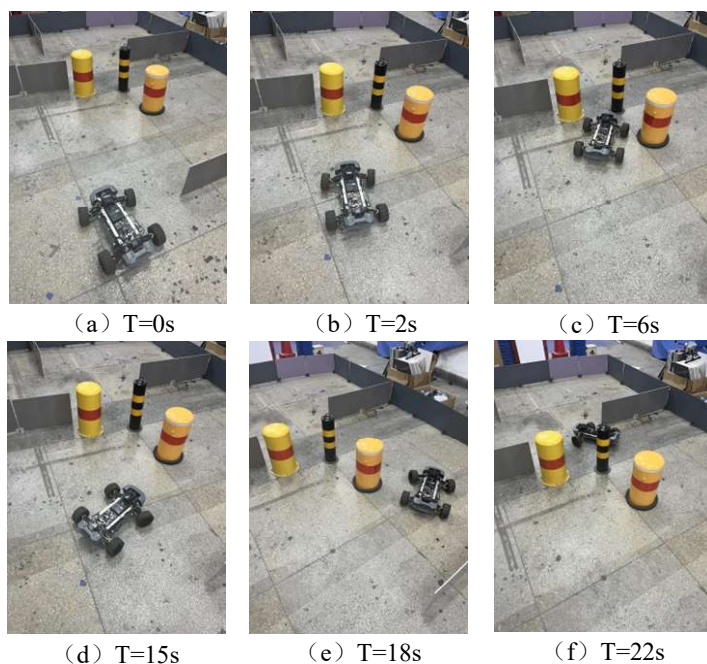


图 5-16 实验过程

移动机器人运行过程中识别障碍物如图 5-17 所示，图 5-17（a）为移动机器人经过障碍物区的初始位置识别 A 障碍物在右方，但是距离过远依然按照全局路径规划的线路行进；图 5-17（b）为移动机器人识别出 B 障碍物，但是距离过远依然按照全局路径规划的线路行进；图 5-17（c）为移动机器人判断无法通过障碍物区，于是选择后退并寻找其他可行的路径继续前进。图 5-17（d）移动机器人后退之后选择从右方绕过。

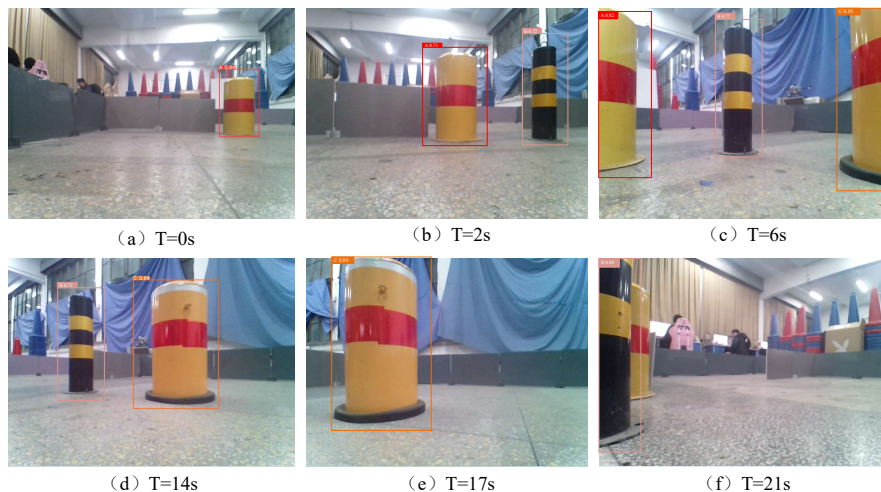


图 5-17 识别过程

在 rviz 中，移动机器人运行的路径轨迹如图 5-18 所示。



图 5-18 移动机器人轨迹

3) 实验结果分析

实验结果表明，在没有临时静态障碍物的环境中，移动机器人将遵循全局路径至目标点。若存在临时障碍物，移动机器人会先接近障碍物寻找绕行路径，

若无法绕过障碍物，不会重复进行无效的方向调整，而是重新规划全局路径，依照新路径前往目标点。

在本实验中结合深度学习与 TEB 算法实现移动机器人的局部避障，TEB 算法通过不断调整移动机器人的位姿来执行避障任务，因此移动机器人在拥挤的环境中可能会进入无法调整位姿的死角区域，导致导航过程失败，如图 5-19 所示。



图 5-19 运动到位姿死区

在存在静态障碍物的地图中，对比只使用 TEB 算法进行导航和深度学习结合 TEB 算法进行导航，如表 5-2 所示。

表 5-2 50 次导航对比

方法	失败（次）	成功（次）	成功率（%）	平均时间（s）
TEB	32	18	36	52
深度学习结合 TEB	14	36	72	40

当障碍物距离较近时，TEB 算法可能无法使移动机器人调整到最佳位姿，因此需要更多时间进行调整位姿，甚至移动机器人的路径规划失败导致停止运动。而使用深度学习与 TEB 算法结合的局部路径规划方法时，移动机器人在面对临时障碍物时表现出更强的适应性，当无法绕过临时障碍物时会选择其他路径进行规避。在表中可看出深度学习与 TEB 算法结合的局部路径规划方法比只使用 TEB 算法的成功率提高了近 40%，平均时间减少了 12s。

5.5 本章小结

本章首先介绍了实验平台整体设计，对硬件和软件系统的模块进行设计。其次在移动机器人上部署 Darknet_ROS，使深度学习和 ROS 进行有效结合并进行 YOLOv5 实时测试。然后将改进前后的路径规划算法在实验室环境下进行实验，验证了改进后的路径规划的实用性。最后在移动机器人上进行了实际环境的测试，在实验场景中新增未知障碍物，结果表明移动机器人能够准确识别出障碍物并进行避障。

第 6 章 总结与展望

6.1 论文的主要工作

本文主要对激光雷达结合 IMU、里程计数据建图、改进全局路径规划的 PRM 算法、改进深度学习中 YOLOv5 网络模型，内容具体如下：

(1) 介绍了移动机器人建图与导航的重要作用，并简述了移动机器人路径规划的国内外研究现状，介绍了一些经典的全局与局部路径规划算法。介绍了深度学习在移动机器人中的发展及研究现状。

(2) 介绍了移动机器人 ROS 操作系统，以及移动机器人坐标系系统模型和阿克曼转向运动学模型。为解决传统激光 SLAM 在地图构建中易出现的定位丢失和不准确问题，使用扩展卡尔曼滤波结合 IMU 和里程计信息构建地图的方法，增加地图构建的成功率。

(3) 以 YOLOv5 深度学习模型为基础，对模型中 Backbone 网络的 C3 进行改进，添加了 CA 注意力机制；并对 YOLOv5 的 NMS 进行改进，使用 DIoU_NMS 提高模型对遮挡物体区域和重叠区域的检测性能；考虑了空间之间的相互作用，引入了递归门控卷积；为了提高对小目标的检测精度，加入了一个小目标检测头。改进后的 YOLOv5 算法在公共数据集训练，经过实验后，改进后的模型训练的 mAP 值为 86.96%，相比于原始 YOLOv5 提升了 2.79%。同时，将本文算法与近几年主流算法进行对比，结果表明，改进后的深度模型能够满足目标识别的实时性。

(4) 对 PRM 算法进行改进，在学习阶段引入人工势场法，从而提高狭窄通道内的采样点数，研究了 TEB 算法与深度学习相结合的局部避障策略。为了适应复杂环境，将全局和局部路径规划结合形成混合路径规划策略。

(5) 从硬件和软件两部分对移动机器人整体进行说明，在移动机器人平台上部署 Darknet_ROS，然后对移动机器人的路径规划和避障进行实验。在实验环境中通过对比可以得到改进的 PRM 算法规划的路径更平滑，并且 TEB 结合深度学习的避障策略可以实时避开新增障碍物。

6.2 下一步研究计划

本课题研究基于深度学习的移动机器人建图与导航技术，由于实验条件限制，下一步的工作展望如下：

（1）激光 SLAM 依赖于传感器数据来构建地图，因而地图精度受到传感器性能的直接影响。为了提高地图构建的准确性，下一步可以考虑将视觉 SLAM 与深度学习技术相结合，以实现实时的语义分割地图构建。

（2）本文主要是在室内进行数据集的拍摄，因此能够识别的障碍物有限，下一步尝试增加环境的多样性和障碍物的种类数量。

（3）全局路径规划的目的在于找到一条从起点通往终点的可行路径，下一步可以进一步优化 PRM 算法，通过根据障碍物的具体分布来选择合适的采样策略。

参考文献

- [1] 潘琪.自主移动机器人的路径规划算法研究[D].湖南大学,2016.
- [2] 华洪,张志安,施振稳,等.动态环境下多重 A*算法的移动机器人路径规划方法[J].计算机工程与应用,2021,57(10):173-180.
- [3] 李旦.基于传感器融合的移动机器人 SLAM 与导航研究[D].安徽工程大学,2022.DOI:10.27763/d.cnki.gahgc.2022.000249.
- [4] 孙九博.动态环境移动机器人安全高效的路径规划方法研究[D].山东大学,2020.DOI:10.27272/d.cnki.gshdu.2020.005498.
- [5] 邹强,丛明,刘冬,等.基于生物认知的移动机器人路径规划方法[J].移动机器人,2018,40(06):894-902.DOI:10.13973/j.cnki.robot.170558.
- [6] 王洪斌,郝策,张平,等.基于 A~*算法和人工势场法的移动机器人路径规划[J].中国机械工程,2019,30(20):2489-2496.
- [7] Sun Y, Fang M, Su Y. AGV path planning based on improved Dijkstra algorithm[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 1746(1): 012052.
- [8] Bohren J, Foote T, Keller J, et al. Little ben: The ben franklin racing team's entry in the 2007 DARPA urban challenge[J]. Journal of Field Robotics, 2008, 25(9): 598-614.
- [9] Bacha A, Bauman C, Faruque R, et al. Odin: Team victortango's entry in the darpa urban challenge[J]. Journal of field Robotics, 2008, 25(8): 467-492.
- [10] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.
- [11] Stentz A. Optimal and efficient path planning for partially-known environments[C]//Proceedings of the 1994 IEEE international conference on robotics and automation. IEEE, 1994: 3310-3317.
- [12] Ferguson D, Stentz A. Using interpolation to improve path planning: The Field D* algorithm[J]. Journal of Field Robotics, 2006, 23(2): 79-101.

- [13] Daniel K, Nash A, Koenig S, et al. Theta*: Any-angle path planning on grids[J]. Journal of Artificial Intelligence Research, 2010, 39: 533-579.
- [14] Likhachev M, Ferguson D, Gordon G, et al. Anytime search in dynamic graphs[J]. Artificial Intelligence, 2008, 172(14): 1613-1643.
- [15] Boor V, Overmars M H, Van Der Stappen A F. The Gaussian sampling strategy for probabilistic roadmap planners[C]//Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C). IEEE, 1999, 2: 1018-1023.
- [16] Hsu D, Jiang T, Reif J, et al. The bridge test for sampling narrow passages with probabilistic roadmap planners[C]//2003 IEEE international conference on robotics and automation (cat. no. 03CH37422). IEEE, 2003, 3: 4420-4426.
- [17] Missiuro P E, Roy N. Adapting probabilistic roadmaps to handle uncertain maps[C]//Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. IEEE, 2006: 1261-1267.
- [18] 程谦,高嵩,曹凯,等.基于 PRM 优化算法的移动机器人路径规划[J].计算机应用与软件,2020,37(12):254-259+296.
- [19] 孙波,姜平,周根荣,等.改进遗传算法在移动机器人路径规划中的应用[J].计算机工程与应用,2019,55(17):162-168.
- [20] 张希闻,肖本贤.基于人工势场法的移动机器人动态路径规划[J].山东农业大学学报(自然科学版),2018,49(06):937-940.
- [21] 孙环.视觉导航下新型车辆避障路径智能规划方法[J].自动化与仪器仪表,2020,(10):39-42.
- [22] Greenhalgh J, Mirmehdi M. Real-time detection and recognition of road traffic signs[J]. IEEE transactions on intelligent transportation systems, 2012, 13(4): 1498-1506.
- [23] Shustanov A, Yakimov P. CNN design for real-time traffic sign recognition[J]. Procedia engineering, 2017, 201: 718-725.
- [24] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE

- p>conference on computer vision and pattern recognition. 2014: 580-587.
- [25] Maldonado-Bascón S, Lafuente-Arroyo S, Gil-Jimenez P, et al. Road-sign detection and recognition based on support vector machines[J]. IEEE transactions on intelligent transportation systems, 2007, 8(2): 264-278.
- [26] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
- [27] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.
- [28] Neubeck A, Van Gool L. Efficient non-maximum suppression[C]//18th international conference on pattern recognition (ICPR'06). IEEE, 2006, 3:850-855.
- [29] 苏丽,孙雨鑫,苑守正.基于深度学习的实例分割研究综述[J].智能系统学报,2022,17(01):16-31.
- [30] 林韩熙,向丹,欧阳剑,等.移动机器人路径规划算法的研究综述[J].计算机工程与应用,2021,57(18):38-48.
- [31] 华丹.基于区块链技术的工业移动机器人视觉检测及避障系统设计[J].计算机测量与控制,2020,28(07):69-73.
- [32] 陈建新,邢琰,李志平,等.祝融号火星车自主环境感知与避障技术[J].中国科学:技术科学,2022,52(08):1186-1197.
- [33] 薛光辉,李瑞雪,张钺昊,等.基于 3D 激光雷达的 SLAM 算法研究现状与发展趋势[J].信息与控制,2023,52(01):18-36.
- [34] Huang A S, Bachrach A, Henry P, et al. Visual odometry and map** for autonomous flight using an RGB-D camera[C]//Robotics Research: The 15th International Symposium ISRR. Springer International Publishing, 2017: 235-252.
- [35] 胡铭超.基于激光 SLAM 的移动机器人地图构建与优化[D].哈尔滨工业大

- 学,2019.
- [36] 盛磊.基于激光雷达的室内移动机器人建图与导航研究[D].武汉理工大学,2018.
- [37] 王随阳,蒋林,雷斌,等.基于多传感器融合的室内移动机器人定位研究[J].组合机床与自动化加工技术,2020,(10):72-75+89.
- [38] 孟含.基于多传感器融合的 AGV 定位算法研究[D].青岛大学,2020.
- [39] 金峰,蔡鹤皋.移动机器人 IMU 与激光扫描测距传感器数据融合[J].移动机器人,2000,(06):470-473.
- [40] 杜哲夫.基于激光雷达的室内移动机器人 SLAM 与导航技术研究[D].湖北工业大学,2019.
- [41] 施俊屹,查富生,孙立宁,等.移动机器人视觉惯性 SLAM 研究进展[J].移动机器人,2020,42(06):734-748.
- [42] 明瑞冬.基于 ROS 的移动机器人自主导航系统研究[D].武汉理工大学,2018.
- [43] 罗海波,许凌云,惠斌,等.基于深度学习的目标跟踪方法研究现状与展望[J].红外与激光工程,2017,46(05):14-20.
- [44] 何春燕.基于卷积神经网络的车行环境多类障碍物检测与识别[D].重庆邮电大学,2017.
- [45] 闫贺,郭秀娟.卷积神经网络在图像处理中的应用[J].吉林建筑大学学报,2021,38(01):84-88.
- [46] Graham B. Fractional max-pooling[J]. arXiv preprint arXiv:1412.6071, 2014.
- [47] Wischik D, Handley M, Braun M B. The resource pooling principle[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(5): 47-52.
- [48] Abadi M, Barham P, Chen J, et al. {TensorFlow}: a system for {Large-Scale} machine learning[C]//12th USENIX symposium on operating systems design and implementation (OSDI 16). 2016: 265-283.
- [49] Gulli A, Pal S. Deep learning with Keras[M]. Packt Publishing Ltd, 2017.
- [50] Neu N, Duchon J, Zachariah P. TORCH infections[J]. Clinics in perinatology, 2015, 42(1): 77-103.

- [51] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the 22nd ACM international conference on Multimedia. 2014: 675-678.
- [52] Biddle P, England P, Peinado M, et al. The darknet and the future of content distribution[C]//ACM Workshop on digital rights management. 2002, 6: 54.
- [53] Chen T, Li M, Li Y, et al. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems[J]. arXiv preprint arXiv:1512.01274, 2015.
- [54] Quinlan S, Khatib O. Elastic bands: Connecting path planning and control[C]//[1993] Proceedings IEEE International Conference on Robotics and Automation. IEEE, 1993: 802-807.
- [55] Rösmann C, Hoffmann F, Bertram T. Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control[C]//2015 european control conference (ECC). IEEE, 2015: 3352-3357.

攻读硕士学位期间主要研究成果

（1）发表的学术论文

1、兰天,欧阳嘉泰,何宇豪,等,改进 Yolov5 框架的交通标志检测算法[J].传感技术学报. (已录)

2、Jiatai OuYang, Yimao Zhao, Tian Lan and Guanling Wang, "Research and design of brushless DC motor cyclic variable speed control system," 2023 35th Chinese Control and Decision Conference(CCDC),Yichang,China,2023,pp.3147-3151,doi:10.1109/CCDC58219.2023.10326874. (已录)

（2）获奖情况

1、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；

2、第十六届全国大学生智能汽车竞赛安徽赛区华为 AI 组三等奖；

3、第十七届全国大学生智能汽车竞赛安徽赛区华为 AI 组一等奖；

4、第十六届中国智能制造挑战赛全国初赛一等奖；

5、第十七届中国智能制造挑战赛全国初赛二等奖；

致 谢

时光荏苒，转眼三年的硕士研究生生活已经接近尾声。这几年的时光既漫长又短暂，其中充满了酸甜苦辣，更有收获和成长，安徽工程大学的校园生活给我留下了美好体验和深刻的回忆。

在此，我要感谢我的研究生导师王冠凌教授，他严谨的工作态度、学术精神和专业素养都对我产生了积极的影响，使我受益终生。在校学习期间，导师不仅毫无保留的指点我学习工作，使我的科研能力和理论水平得到了很大的提升，在生活上也给予了很大的关心和照顾。

此外，我也要感谢陪伴我度过研究生生涯的师兄师弟，我们在生活中互相关照，在学习和工作中互相督促，结下了深厚的友谊，我深深的希望在毕业之后我们还能维系这份情谊。最后，感谢我的父母这么多年来对我的培养与爱护，你们的支持是我进步最大的动力！

尚德致学 唯实惟新

