2025 **年** 6**月** 1 **日**

目 录

[1 引言 1](#_Toc207534096)

[1.1 研究背景及意义 1](#_Toc207534097)

[1.2 国内外研究现状 1](#_Toc207534098)

[1.3 研究内容 2](#_Toc207534099)

[2 系统分析 4](#_Toc207534100)

[2.1 可行性分析 4](#_Toc207534101)

[2.1.1 技术可行性分析 4](#_Toc207534102)

[2.1.2 经济可行性分析 4](#_Toc207534103)

[2.2 系统需求分析 5](#_Toc207534104)

[2.2.1 功能性需求分析 5](#_Toc207534105)

[2.2.2 非功能性需求分析 6](#_Toc207534106)

[2.3 系统用例分析 6](#_Toc207534107)

[3 系统设计 8](#_Toc207534108)

[3.1 总体设计 8](#_Toc207534109)

[3.2 数据库设计 8](#_Toc207534110)

[4 系统详细设计与实现 12](#_Toc207534111)

[4.1 服务器端 12](#_Toc207534112)

[4.1.1 基于Qwen的对话寻车界面 12](#_Toc207534113)

[4.1.2 CPR设备管理 13](#_Toc207534114)

[4.1.3 CPR识别信息管理 15](#_Toc207534115)

[4.1.4 点位信息管理 18](#_Toc207534116)

[4.1.5 机器人管理 22](#_Toc207534117)

[4.1.6 用户管理及管理平台登录认证功能 24](#_Toc207534118)

[4.2 CPR设备端 26](#_Toc207534119)

[4.2.1 基于YOLOv11和Paddle OCR的车牌-车位识别匹配功能 27](#_Toc207534120)

[4.3 机器人端 31](#_Toc207534121)

[4.3.1 基于ROS的机器人控制算法 31](#_Toc207534122)

[4.3.2 基于Bottle WEBUI的状态显示界面 38](#_Toc207534123)

[5 系统测试 41](#_Toc207534124)

[5.1 测试目的 41](#_Toc207534125)

[5.2 功能测试 41](#_Toc207534126)

[5.3 测试结论 43](#_Toc207534127)

[6 总结与展望 44](#_Toc207534128)

[6.1 本文工作总结 44](#_Toc207534129)

[6.2 存在不足及未来展望 44](#_Toc207534130)

[参考文献 45](#_Toc207534131)

# 引言

## 研究背景及意义

随着城市化进程持续不断地向前推进，机动车的保有量急剧增加，寻车难这一问题已然成为困扰城市居民的普遍且棘手的难题，在一些如大型商业中心、医院以及交通枢纽等地方，停车场的面积较为广阔，车位数量众多，然而车主却时常遭遇寻找停车位置方面的困扰。依据相关调查显示，车主一般需要耗费15至20分钟的时间去寻找自己停放的车辆。这样做白白浪费了宝贵的时间，而且还容易引发车主的焦虑情绪，对出行体验造成严重的不良影响。

在此背景下，停车场寻车机器人呈现出了巨大的应用潜力以及实际价值，机器人可凭借高效的车辆定位以及路径引导服务，协助车主快速找到停车位置，有效地减少寻找车辆所花费的时间，提高停车效率并提升用户满意度，寻车机器人的广泛应用还可使停车场管理流程得到优化，降低人工成本，提高运营效益，对改善城市交通管理、缓解交通压力有着关键的推动作用。

即便如此，目前市场上还没有成熟的停车场寻车机器人产品，该领域仍然处在技术探索以及发展的阶段。当前停车场的环境复杂且变化多端，光线昏暗、信号干扰比较严重，这给机器人的定位与导航能力给予了严峻的挑战。为了保证机器人可在这种复杂环境中精准地定位车辆，它要拥有高精度的传感器和导航技术。另外停车场内的人员和车辆动态变化频繁，机器人还需要有强大的避障能力以及实时路径规划功能，保证在任何时候可安全、高效地引导车主。

寻车机器人要和停车场管理系统实现深度集成，达成数据共享并协同工作，这对系统整体集成性与兼容性有了更高要求，当下市场上成熟寻车机器人产品匮乏，本研究就针对这些技术挑战展开探讨，借助设计智能化、自动化的停车场寻车机器人系统，填补市场空白，帮助解决城市停车难题，推动智慧停车行业快速发展。

## 国内外研究现状

当下智能寻车系统的研究已然成为智慧城市建设里的关键构成部分，随着城市化进程的加速以及汽车保有量的不断增长，寻找车辆困难已成为停车场管理系统中的焦点问题，现有的技术方案于多个关键环节依旧存有不足之处，这些研究主要聚焦于车位检测、路径规划以及移动应用开发等关键技术环节，然而在系统集成度与智能化水平方面仍有较大的提升空间。

在车位检测与车辆定位技术方面，国内外学者提出了多种技术路线，各具特色但也存在明显局限。国内李月等人开发的物联网停车系统采用RFID（Radio Frequency Identification）技术进行车辆定位，该系统通过布设大量RFID读写器实现了0.5米精度的定位，测试显示其定位误差可控制在0.5米范围内。然而，该系统在多车并排停放的高密度场景下，由于信号干扰和标签碰撞问题，识别率会显著下降至65%以下[1]。国外Allah Ditta等人提出的ANPR系统采用了深度学习算法实现车牌识别，在理想光照条件下识别准确率可达95%的较高水平。但该系统在逆光、雨雾等复杂天气条件下的鲁棒性不足，性能会大幅降低至70%以下[2]。此外，幸敏团队提出的基于ZigBee组网的超声波测距方案虽然实现了98%的空位识别准确率，但单节点硬件成本超过200元，限制了其大规模商业应用[3]。

在用户交互设计方面，现有系统普遍存在体验不佳、操作复杂等问题。国内陈黎明等开发的微信小程序寻车系统虽然降低了用户使用门槛，但操作繁琐，在大型停车场中这一操作过程往往需要1-2分钟，用户体验评分仅为2.8/5分[4]。国外Ruchi Rani团队提出的IPS系统创新性地集成了二维码支付功能，简化了缴费流程，但系统缺乏实时导航指引功能，用户仍需自行在复杂的停车场环境中寻找车辆，未能从根本上解决寻车难题[5]。麻吉辉等开发的PC客户端虽然提供了多种查询方式，但需要用户在固定终端操作，灵活性不足[6]。

值得注意的是，通过对现有文献的全面梳理发现，目前国内外尚未出现成熟的机器人寻车系统解决方案。现有方案多依赖于固定式终端或用户手机等被动式服务模式，无法实现主动服务。国外学者Mert Ozkaya提出的参考架构虽然从理论上考虑了三维环境建模等问题，构建了包含上下文、模块、组件和分配等多个视角的系统框架，但仍停留在概念设计阶段，缺乏实际应用验证[7]。Dinesh Sahu等提出的数字孪生框架虽然实现了停车场的虚拟建模，但尚未与实体机器人平台结合[8]。国内周伟在多AGV系统调度方面的研究为机器人协同控制提供了重要参考，但主要针对车辆存取场景，未涉及寻车服务[9]。

在系统集成方面，现有技术方案普遍存在兼容性差、扩展困难等问题。国内李今团队开发的基于STM32的停车管理系统采用蓝牙和WiFi双模通信，虽然实现了车位状态的秒级更新，但由于采用私有通信协议，难以与其他系统互联互通[10]。国外Shreeram Hudda提出的边缘计算框架虽然通过R-CNN（Region-CNN）模型将车位状态误报率控制在5%以下，但其系统架构与主流物联网平台存在兼容性问题[11]。孔德财等的研究指出，当前智能停车系统存在"信息孤岛"现象，不同系统间的数据共享和功能协同亟待加强[12]。

综上所述，当前智能寻车技术在多个关键环节仍存在明显不足：在车牌识别方面，现有方案的环境适应性有待提升；在用户交互方面，操作便捷性和导航精准度需要改进；在系统载体方面，缺乏自主移动的机器人平台；在系统集成方面，兼容性和扩展性有待加强。这些技术瓶颈严重制约了智能寻车系统的实际应用效果和用户体验，亟需新的技术方案来解决这些问题。未来研究需要在这些方面进行重点突破，以推动智能寻车技术的实际应用和产业化发展。

## 研究内容

本文聚焦于基于 SLAM（Simultaneous Localization and Mapping，即时定位与地图构建） 技术的停车场智能寻车机器人系统的设计与实现工作，期望能提升停车场寻车效率并优化用户体验，具体研究内容如下：

（1）服务器端功能：服务器端作为系统的核心管理平台，承担着处理用户请求、开展任务调度以及进行数据存储与管理等工作。其主要功能有：基于 Qwen 大语言模型构建的对话寻车界面，该界面支持语音与文本输入，让用户可依靠语音或者文本与系统进行交互来查询车辆信息，管理员功能模块覆盖 CPR（Car Plate Recongnising）设备管理、车牌识别信息管理、机器人管理、点位信息管理以及用户信息管理等，可实现对系统硬件设备、机器人状态以及用户信息的实时管理。还有车辆查询、路径规划以及任务调度等功能，以此保证系统能及时响应用户需求并优化寻车流程。

（2）CPR 设备端功能：CPR 设备端主要负责车牌识别以及停车位状态监测，它借助安装在停车场内的摄像头采集车位图像，借助 YOLOv11 目标检测算法与 Paddle OCR 来进行车牌识别，随后将识别结果上传至服务器。其核心功能包括：基于 YOLOv11 和 Paddle OCR 的车牌 - 车位识别匹配功能，可精准识别车牌并与停车位进行匹配，实时更新停车位的占用状态，可支持多个车位的实时监控以及数据传输，保障停车场信息及时更新且准确。

（3）机器人端功能：机器人端是系统的执行平台，主要依据用户的查询请求引导车主找到停车位并完成寻车任务。其主要功能有：利用 SLAM 技术进行自主导航与定位，借助激光雷达和深度相机进行环境感知，生成停车场的实时地图，进行路径规划和避障，实现两种寻车模式，即定点扫码寻车和非定点扫码寻车，在定点扫码寻车模式下，车主扫描指定二维码后，机器人引导车主到达车辆位置，在非定点扫码寻车模式下，车主扫描机器人上的二维码后，机器人带领车主到指定车辆位置。机器人有较强的避障能力以及稳定的路径规划功能，可应对动态变化的停车场环境，保障车主安全顺利寻车。

借助上述三个端的设计与实现，本系统可有效地支持停车场内的车辆定位和导航任务，提供便捷的用户交互界面，实现车辆识别与路径规划的自动化，提高停车场的运作效率和用户体验。

# 系统分析

## 可行性分析

为保障本项目能够顺利进行，本节会从技术可行性以及经济可行性这两方面着手，对项目展开分析。

### 技术可行性分析

就技术可行性而言，本项目所依托的技术根基相对成熟，拥有可行性。所运用的编程语言Python在机器人控制、数据处理以及系统集成等方面有着广泛应用，可保证系统开发稳定运行。针对车牌识别，本项目采用YOLOv11目标检测算法以及百度Paddle OCR来进行车牌识别，并且已在多种复杂环境中获得验证，可以有效解决车牌识别问题；ROS框架在机器人控制领域被广泛运用，可以实现精准的导航以及任务执行。另外SLAM技术依靠实时构建停车场地图并实施定位，可应对停车场内复杂环境的变化，保障机器人稳定运行。这些技术经过大量应用与验证，已然足够成熟，可支撑本项目顺利实施。

项目的技术方案经过合理设计，保证了系统的各个模块可高效集成，借助MySQL数据库进行数据管理与存储，搭配Bottle框架开发后台系统，可实现实时数据更新与交互，保证车主和停车场管理员可便捷地获取信息并操作系统。ROS框架为机器人提供了稳定的控制平台，集成多种传感器和执行器，使机器人可在复杂的停车场环境中完成任务。整个系统架构可实现高效的任务调度以及实时数据处理。从技术层面来讲，本项目有充足的可行性。

### 经济可行性分析

从经济可行性方面来说，项目的初期投入主要覆盖硬件设备、软件开发以及系统集成的费用，各项硬件成本价格如表3-1所示。硬件成本主要集中在传感器、机器人平台以及计算单元的采购上，尽管初期投入较大，然而设备使用寿命较长，并且随着规模化应用，成本会逐渐降低，在软件开发和系统集成方面，运用成熟的开源技术平台，避免了高额许可费用，节约了开发成本，运营和维护费用较为可控，主要涉及系统维护和人工成本。

**表3-1 硬件成本价格表**

**Tab. 3-1 The table of hardware cost price**

|  |  |
| --- | --- |
| 设备名称 | 成本价格 |
| 机器人设备 | 4499元 |
| CPR设备 | 769元 |
| 服务器设备 | 3269元 |

## 系统需求分析

### 功能性需求分析

#### 后台管理系统功能

（1）系统需要支持管理员登录和身份认证功能，管理员可以通过用户名和密码登录后台管理系统，确保只有授权用户可以访问。

（2）后台管理系统应包括多个模块：CPR设备管理、CPR识别信息管理、点位信息管理、机器人信息管理、用户信息管理和管理平台登录认证功能。

（3）CPR设备需要能够使管理员进行CPR设备的添加、修改和删除操作，实时管理设备的状态。

（4）CPR识别信息管理功能应支持设置、查看和编辑车牌识别区域以及与车位的对应关系。

（5）点位信息管理功能应允许管理员添加、修改、删除点位信息，并能基于机器人位置进行点位回传。

（6）机器人信息管理功能应包括对机器人的添加、删除和状态修改（例如服务模式、暂停服务、调试模式等）的管理。

（7）用户信息管理功能应支持添加、编辑和删除用户信息，专门针对管理员进行管理。

#### 用户界面和交互功能

（1）用户（车主）应通过前台页面进行交互，界面需要包括基于Qwen的语音交互功能，允许用户通过语音或文本输入查询车辆信息。

（2）系统应根据车主的需求提供寻车服务，用户通过语音或文本查询输入车牌号，系统能够识别并返回停车位信息。

（3）用户界面应支持基于YOLOv11和Paddle OCR的车牌识别功能，确保车主能快速查询到车辆所在位置。

（4）前端界面需要展示基于ROS的机器人控制算法，包括机器人如何引导车主找到停车位。

（5）系统应集成基于Edge-TTS的语音合成功能，提供语音播报功能，以引导用户进行寻车。

#### 机器人控制和路径规划

（1）机器人端需要支持两种寻车模式：定点扫码寻车模式和非定点扫码寻车模式。

（2）在定点扫码寻车模式下，车主通过扫描二维码，系统会为其分配最近的机器人，引导其到车辆所在的位置。

（3）在非定点扫码寻车模式下，车主扫描机器人上的二维码后，机器人将带领车主到达目标车辆。

（4）机器人应具备良好的避障能力，能够在停车场内应对动态环境中的障碍物，并安全引导车主。

### 非功能性需求分析

（1）性能需求方面，系统需有高效的数据处理能力和较快的响应速度，车辆查询和路径规划的响应时间应小于2秒，保证车主能迅速获得车辆定位信息和导航指引。系统应能在高并发情况下正常运行，支持多个车主同时进行寻车操作且不影响系统性能，

（2）可靠性和稳定性方面，系统应有较高的可靠性和稳定性，可在停车场环境中长时间运行而不出现崩溃或故障，系统在处理车主请求、定位查询、路径规划等任务时，应能稳定高效地响应，避免出现延迟或错误，保证车主的使用体验。

（3）安全性方面，系统应保证数据的安全，保护车主和停车场管理者的敏感信息，保证信息不会被未授权用户获取或篡改，另外系统对机器人与车主的交互过程进行安全控制，防止机器人在执行任务时发生意外事故。

## 系统用例分析

本系统用户（车主）用例图如图3-1所示。

图片包含 图示

AI 生成的内容可能不正确。

**图3-1 用户用例图**

**Fig. 3-1 The use case diagram of user**

本系统管理员用例图如图3-2所示。

图示

AI 生成的内容可能不正确。

**图3-2 管理员用例图**

**Fig. 3-2 The use case diagram of administrator**

# 系统设计

## 总体设计

本系统以智能寻车服务为核心目标，采用模块化分层架构，整合服务器端、CPR（Car Plate Recognizing）端与机器人端三部分协同工作。系统整体实现过程始于用户交互层，用户通过语音或文本输入寻车请求，服务器端基于Qwen大模型解析语义并识别意图，调用机器人调度功能。CPR设备通过摄像头实时采集车位图像，结合YOLOv11模型检测车牌位置，并利用Paddle OCR完成文字识别，将车位状态至服务器数据库。机器人端利用ROS框架实现机器人导航控制，并设计两种寻车方式：定点扫码寻车与非定点扫码寻车两种寻车模式。

图4-1为系统功能模块图。

图示

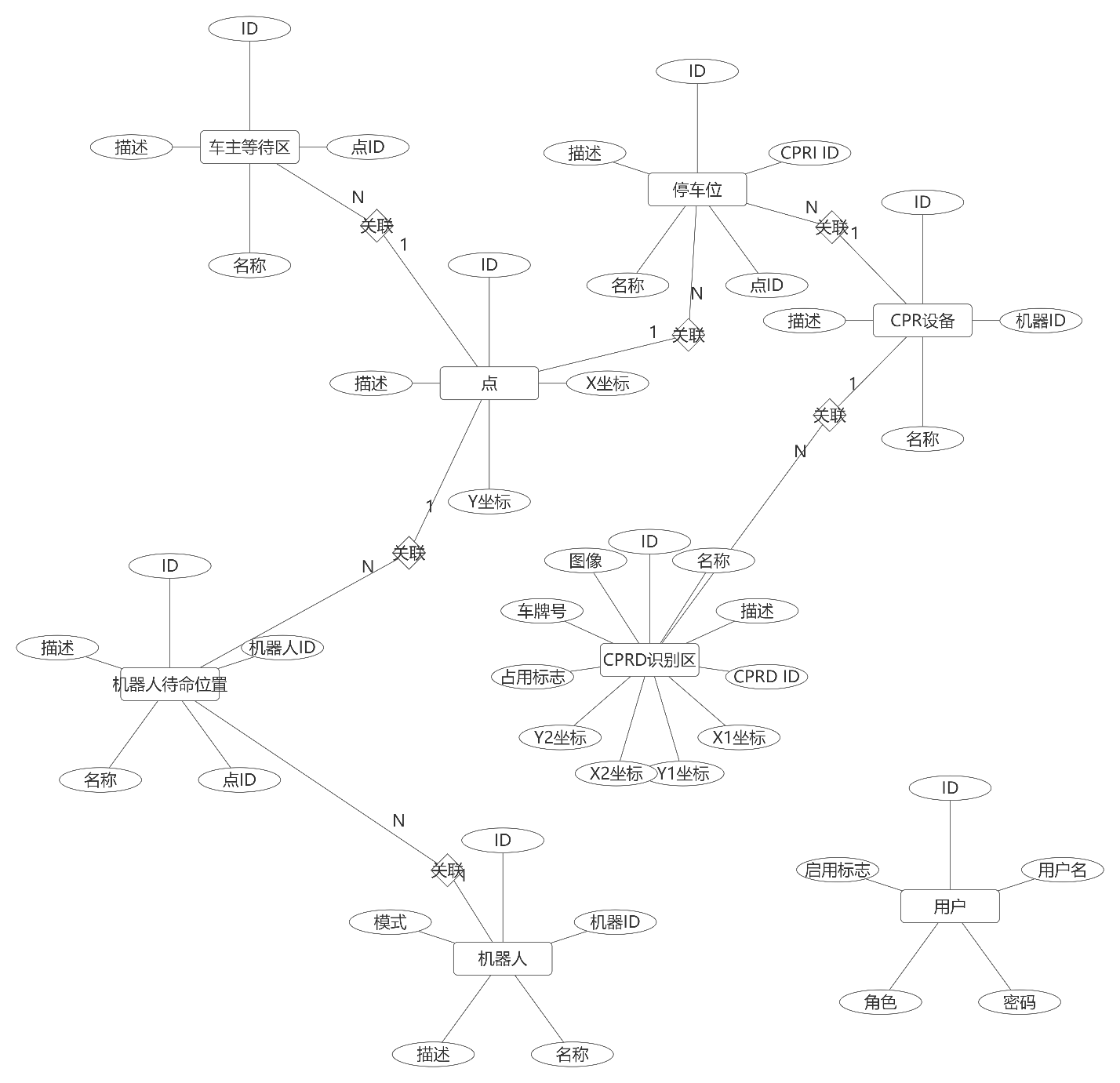
AI 生成的内容可能不正确。

**图4-1 系统功能模块图**

**Fig. 4-1 The module diagram of system function**

## 数据库设计

为了实现数据的存储，本项目使用MySQL数据库进行数据存储。其中，总体数据库E-R图如下图4-2所示。



**图4-2 数据库E-R图**

**Fig. 4-2 The E-R diagram of database**

本项目数据库共有8个数据表，各表的属性信息如下表所示。

**表4-1 车主等待区数据表**

**Tab. 4-1 The table of car owner waiting area data**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 说明 |
| id | int | 11 | ID |
| point\_id | int | 11 | 点ID |
| name | varchar | 255 | 名称 |
| description | text | - | 描述 |

**表4-2 CPR设备数据表**

**Tab. 4-2** **The table of CPR device data**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 说明 |
| id | int | 11 | ID |
| machine\_id | varchar | 255 | 机器ID |
| name | varchar | 255 | 名称 |
| description | text | - | 描述 |

**表4-3 CPR识别区域数据表**

**Tab. 4-3 The table of CPR recognition area data**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 说明 |
| id | int | 11 | ID |
| name | varchar | 255 | 名称 |
| description | text | - | 描述 |
| cprd\_id | int | 11 | CPR ID |
| x1 | float | - | X1坐标 |
| y1 | float | - | Y1坐标 |
| x2 | float | - | X2坐标 |
| y2 | float | - | Y2坐标 |
| occupying\_flag | int | 1 | 占用标志 |
| car\_plate\_no | varchar | 255 | 车牌号 |
| image | longblob | - | 图像 |

**表4-4 停车位数据表**

**Tab. 4-4 The table of parking space data**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 说明 |
| id | int | 11 | ID |
| cpri\_id | int | 11 | CPRI ID |
| point\_id | int | 11 | 点ID |
| name | varchar | 255 | 名称 |
| description | text | - | 描述 |

**表4-5 点位数据表**

**Tab. 4-5 The table of point data**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 说明 |
| id | int | 11 | ID |
| x | double | - | X坐标 |
| y | double | - | Y坐标 |
| description | text | - | 描述 |

**表4-6 机器人数据表**

**Tab. 4-6 The table of robot data**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 说明 |
| id | int | 11 | ID |
| machine\_id | varchar | 255 | 机器ID |
| name | varchar | 255 | 名称 |
| description | text | - | 描述 |
| mode | int | 11 | 模式 |

**表4-7 机器人待机位置数据表**

**Tab. 4-7 The table of robot standby position data**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 说明 |
| id | int | 11 | ID |
| robot\_id | int | 11 | 机器人ID |
| point\_id | int | 11 | 点ID |
| name | varchar | 255 | 名称 |
| description | text | - | 描述 |

**表4-8 用户数据表**

**Tab. 4-8 The table of user data**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 说明 |
| id | int | 11 | ID |
| username | varchar | 255 | 用户名 |
| password | varchar | 255 | 密码 |
| role | int | 11 | 角色 |
| enable | int | 11 | 启用标志 |

# 系统详细设计与实现

## 服务器端

### 基于Qwen的对话寻车界面

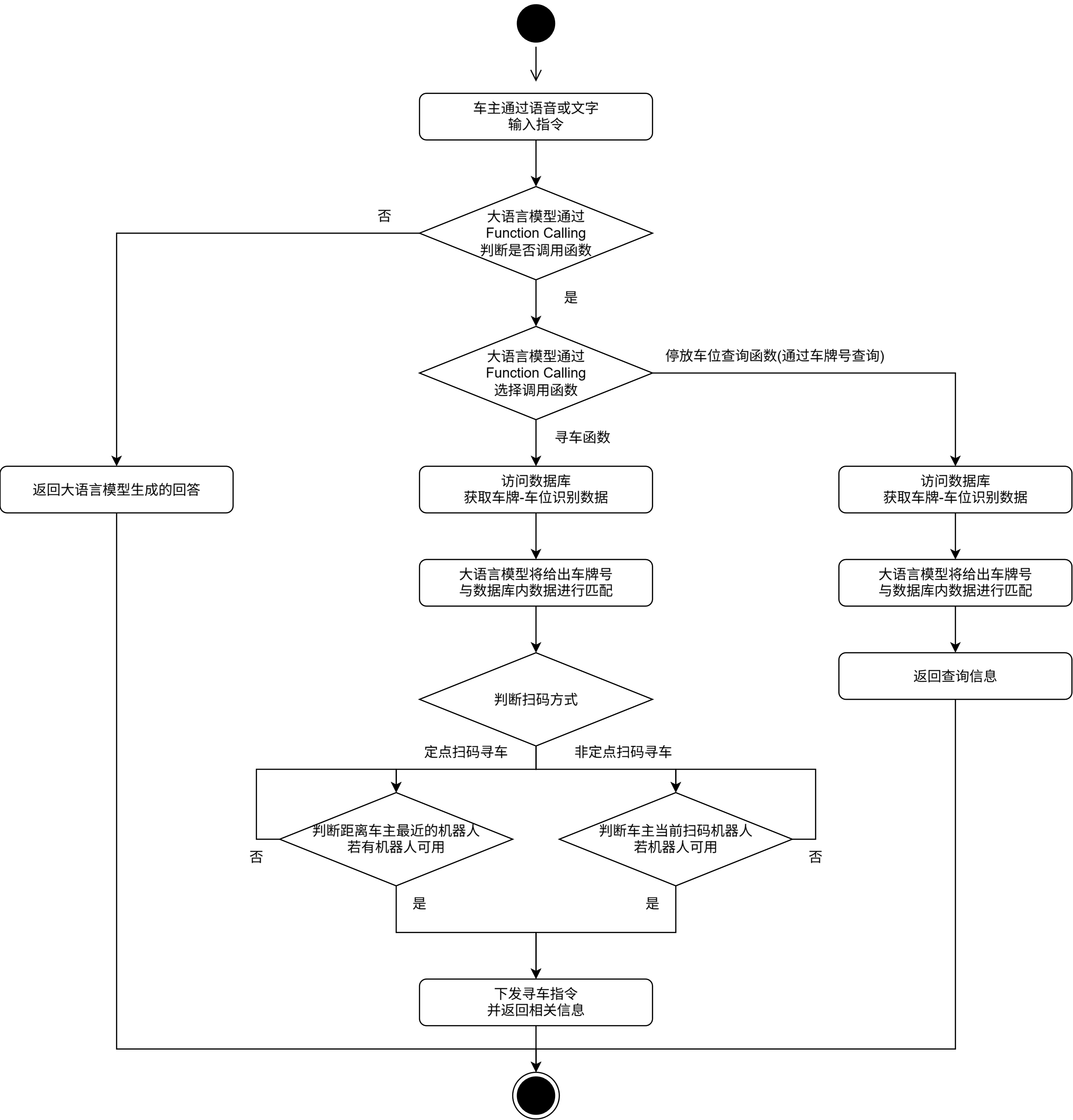
本项目的对话寻车界面采用Qwen大模型，支持语音与文本两种交互模式。用户通过前端输入框提交文字查询或通过麦克风发起语音指令。若用户选择语音输入，系统将用户语音上传至百度短语音识别平台进行识别，将音频转化为文本数据。交互过程中，系统依托大语言模型完成语义解析与意图识别，并同时判断是否需要使用Function Call调用项目中的相关函数。Funtion Call可调用的函数包含了寻车函数和停放车位查询函数。最后，大模型给出回复，并调用基于Edge-TTS的语音合成功能合成语音进行播放。



**图5-1基于Qwen的对话寻车界面图**

**Fig. 5-1 The Interface of dialogue car search based on Qwen**

功能调用机制包含两种核心操作：车辆位置检索与寻车服务调度。车辆检索模块通过车牌信息与数据库进行结构化数据匹配，输出精准停车位置信息。寻车服务模块则根据用户扫码方式实施差异化处理：选择定点扫码寻车模式时匹配最近可用机器人，若资源不足则启动排队等候机制；选择非定点扫码寻车时直接指定当前扫码机器人，进行导航任务。



**图5-2基于Qwen的对话寻车流程图**

**Fig. 5-2 The flowchart of dialogue car search based on Qwen**

### CPR设备管理

CPR设备管理功能是针对系统接入的CPR设备开展统一管理工作的，借助此管理功能，管理员可便利地实施设备的添加、修改、删除以及浏览等操作，以此保证设备可以高效且稳定地提供车牌识别服务，设备管理功能的达成涉及到设备接入、设备信息管理以及与设备进行交互的功能模块。

#### 设备添加与注册

CPR设备接入时首先要依靠网络和服务器建立连接，设备连接之后，管理员利用系统提供的设备添加界面来进行设备注册，添加设备之际，管理员需输入设备的基本信息，比如机器ID、设备名称以及描述等，设备添加过程包含两个关键步骤，即设备的连接检测与设备信息的存储。CPR设备添加界面如图5-3所示。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-3 CPR设备管理-设备添加界面**

**Fig. 5-3 The interface of devices addition in CPR devices management**

设备添加成功后，系统会把该设备的信息存入数据库，并且和设备的网络连接进行映射，管理员可在设备列表页面查看所有注册设备的状态以及信息，对设备开展后续的管理操作。

#### 设备修改

设备信息修改功能让管理员可以更新已注册设备的基本信息，像设备名称、描述等，管理员可借助设备修改界面输入新的设备信息，系统会经由数据库更新相应的设备记录。修改操作同时会验证设备ID的正确性，保证管理员只能修改已存在的设备信息。CPR设备修改界面如图5-4所示。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-4 CPR设备管理-设备修改界面**

**Fig. 5-4 The interface of devices modification in CPR devices management**

#### 设备删除

设备删除功能给予管理员清除不再使用设备的能力，借助删除设备接口，管理员可依据设备ID删除系统中的设备记录，并且同步移除设备与网络连接的映射，删除操作执行时会先进行确认，以防止误操作。

#### 设备状态与连接管理

除了基本的设备管理功能之外，系统还实现了对设备的实时状态监控，每个设备在系统里都有对应的网络连接状态，当设备正常连接到服务器时，管理员可在设备列表中看到设备状态和机器ID等信息，系统借助实时监控和映射，保证设备状态的准确性。CPR设备列表界面如图5-5所示。

图形用户界面, 应用程序, 表格, Excel

AI 生成的内容可能不正确。

**图5-5 CPR设备管理-设备列表界面**

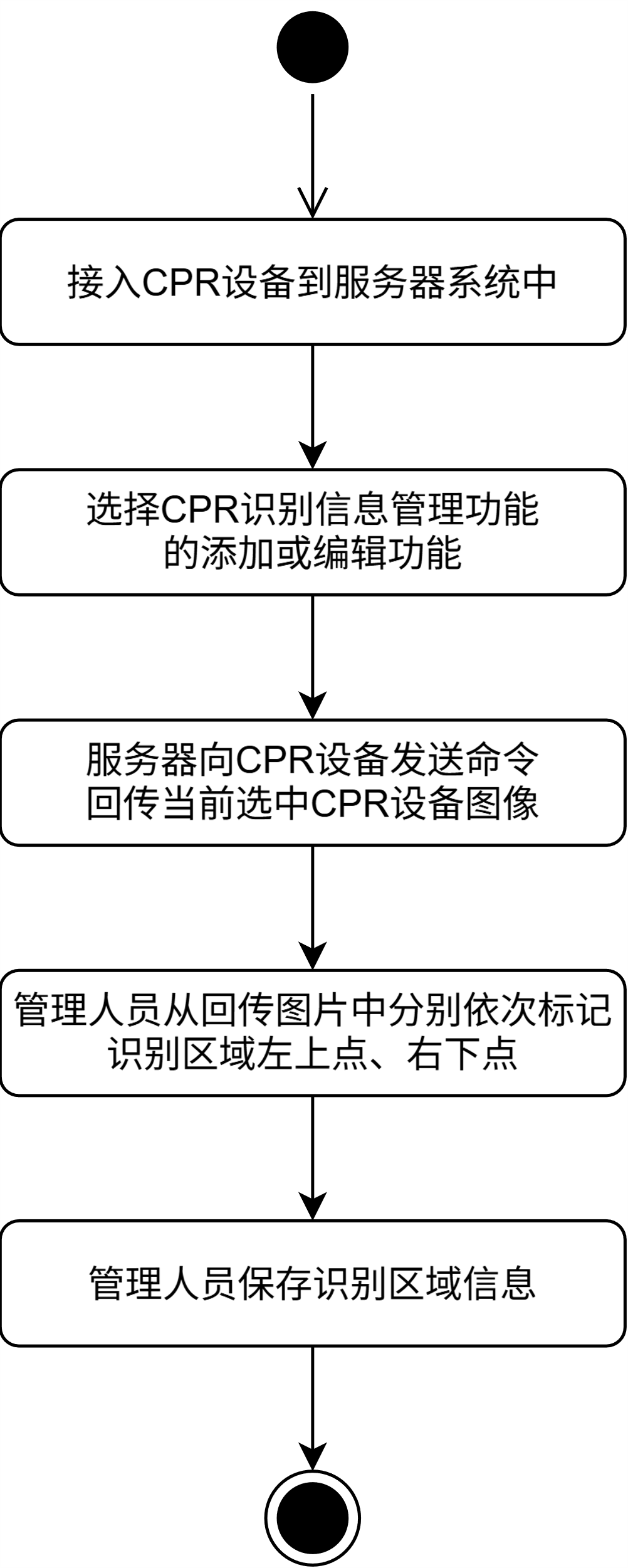
**Fig. 5-5 The interface of devices list in CPR devices management**

### CPR识别信息管理

CPR识别信息管理功能实现了设定识别区域、查看识别区域停放车辆信息两项功能。

在管理人员接入CPR设备至系统中后，可通过本功能配置每台CPR设备对应的CPR识别区域（一台CPR设备可以对应多个识别区域，取决于摄像头视野范围）。识别区域可根据车位在摄像头中占用的区域进行设置。当有车牌出现在识别区域时，CPR设备将识别识别区域内的车牌，从而实现车牌识别。

为了简化识别区域的配置流程，本功能提供了一种基于图像标记识别区域的方法（如图5-6所示）。首先，管理人员需通过CPR设备管理将CPR设备接入到当前系统。随后，管理人员需要选择CPR识别信息管理的添加或编辑选项。进入添加或编辑选项后，系统会向选定CPR设备发送请求命令，CPR设备回传设备图像。管理人员从回传的图像中分别依次选择预设置的识别区域的左上点和右下点。最后，管理人员保存当前设置，识别区域即设置完成。



**图5-6 基于图像标记识别区域的方法流程图**

**Fig. 5-6 The flowchart of region identification method based on image labeling**

#### 识别信息列表

CPR识别信息列表界面如图5-7所示。在列表界面里，借助Bottle后端所渲染的HTML表格来呈现所有CPR信息，每一条CPR信息涉及ID、区域名称、描述、绑定的CPR设备信息、车牌号、占用状态以及操作按钮，操作按钮能让用户开展编辑、查看详情或者删除操作，数据经由后端传递给模板，利用for循环把每一条CPR信息展示在表格的行当中。占用状态依据occupying\_flag的判断来决定显示“空闲”还是“占用”，并为每个CPR增添相应的链接，指向编辑页面、详情页面或者删除操作，删除操作凭借调用API达成，直接从数据库里删除相应记录。

图形用户界面, 文本

AI 生成的内容可能不正确。

**图5-7 CPR识别信息管理-识别信息列表界面**

**Fig. 5-7 The interface of** **recognition information list in CPR recognition information**

#### 识别信息添加

CPR识别信息添加界面如图5-8所示。在添加界面中，表单接收区域名称、描述、绑定的CPR设备，还允许用户凭借点击图像去标记坐标。图片上的坐标经JavaScript捕捉后转换成图片原始大小的坐标，随后显示在页面的输入框内，为了达成图像的坐标标记，JavaScript监听图片容器的点击事件，计算点击位置的坐标，并依据缩放比例转换为原始图像的坐标，点击第一次时，保存x1和y1坐标，第二次点击时保存x2和y2坐标，并且在图片上标记坐标点。表单提交时，所有数据被发送到后端，后端存储这些数据，并在成功添加时返回一条提示信息。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-8 CPR识别信息管理-识别信息添加界面**

**Fig. 5-8 The interface of** **recognition information addition in CPR recognition information**

#### 识别信息修改

CPR识别信息修改界面如图5-9所示。在修改界面中，页面呈现当前CPR信息的详细内容，并允许用户修改部分字段，绑定的CPR设备是一个下拉框，用户可选择当前在线的设备，设备信息依靠后端传递到前端。与添加界面相似，用户可以点击图片来更新坐标，而且修改的坐标值会更新到表单中，当用户提交修改时，表单数据将凭借GET请求提交到后端，后端根据传入的ID找到相应的记录并更新其内容，成功后返回到列表页面并显示修改成功的提示。

图形用户界面, 应用程序, PowerPoint

AI 生成的内容可能不正确。

**图5-9 CPR识别信息管理-识别信息修改界面**

**Fig. 5-9 The interface of** **recognition information modification in CPR recognition information**

#### 识别信息查看

CPR识别信息查看界面如图5-10所示。在信息查看界面中，用户不能修改任何信息，仅能查看当前CPR记录的详细内容。该页面展示CPR的ID、区域名称、描述、占用状态、车牌号、图片等信息，图片以Base64编码的形式进行展示，如此便不需要额外的文件请求，直接借助HTML中的<img>标签加载，占用状态借助检查occupying\_flag来判断，显示相应的状态文本，当占用状态为“占用”时，显示车牌号，否则不显示。在该页面用户只能查看这些信息，不允许修改，并且有一个返回按钮可返回到上一页面。

**隐私因素该图片不予展示**

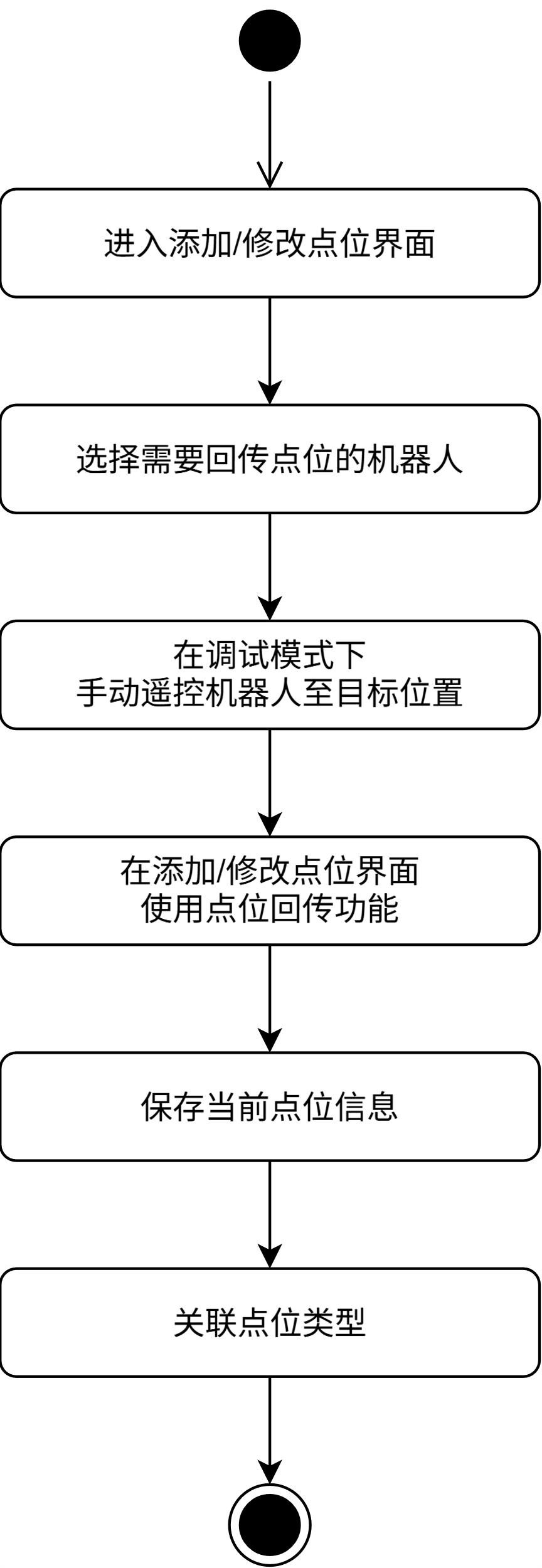
**图5-10 CPR识别信息管理-识别信息查看界面**

**Fig. 5-11 The interface of recognition information view in CPR recognition information**

### 点位信息管理

点位信息管理实现了为机器人导航提供目标点的坐标信息，并通过绑定点位类型与CPR信息、机器人待机点位信息、车位信息等进行关联。从而实现数据之间的关联。

为了解决点位添加/修改功能中X、Y坐标无法可视化添加的问题，本功能提供了一种基于机器人位置的点位回传功能。需添加点位时，管理人员需进入添加/修改点位界面，选择欲回传点位的机器人（机器人必须处于调试模式状态下）。随后，通过遥控器控制机器人抵达目标位置，从界面中使用点位回传功能，将选中机器人的点位自动填充到界面的表单中，从而实现了点位的半自动化导入。最后，管理人员需要保存当前点位，并根据需要将点位与点位类型进行绑定。基于机器人位置的点位回传功能的流程图如图5-11所示。



**图5-11 基于机器人位置的点位回传功能流程图**

**Fig. 5-11 The flowchart of point location feedback function based on robot position**

#### 点位列表

点位列表界面如图5-12所示。在点位列表界面里，系统运用一个表格来呈现所有的点位信息，借助调用数据库或者数据存储服务，获取点位数据，而后把它动态渲染至页面上，每个点位展示其ID、X坐标、Y坐标以及描述信息，并且还设有操作按钮，准许用户进行编辑和删除操作，当用户点击编辑按钮时，系统会跳转至点位修改界面，传递该点位的ID当作查询参数，删除操作会经由发送DELETE请求或者API接口来移除对应的点位，接着刷新列表页面。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-12 点位信息管理-点位列表界面**

**Fig. 5-12 The interface of point list in point information management**

#### 点位添加

点位添加界面如图5-13所示。在添加界面当中，管理人员可选择调试模式下的机器人，依靠点击“回传当前选定机器人实时点位”按钮，触发前端的AJAX请求，此请求会调用服务器的API获取选中机器人的实时位置，利用jQuery和AJAX，系统向后端发起请求，请求参数是机器人的ID，后端收到请求后查询机器人的位置并返回JSON格式的坐标数据。成功获取数据后，前端会把返回的坐标值填入X和Y输入框中，让管理人员可查看并修改这些值，另外管理员还得填写描述信息，最终借助表单提交接口，将新点位的数据保存到数据库中，此部分采用POST请求来提交数据，提交成功后，系统会返回一个成功的提示，并重定向到点位列表页面。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-13 点位信息管理-点位添加界面**

**Fig. 5-13 The interface of point addition in point information management**

#### 点位修改

点位修改界面如图5-14所示。在修改界面中，用户先是凭借点击列表界面的“编辑”按钮进入，系统会依据传递的点位ID从后端加载该点位的详细信息，包含坐标和描述，页面会自动在表单字段中填充这些信息，管理员可查看和修改这些内容，和添加界面类似，修改界面也有回传机器人位置的功能，依靠AJAX请求向后端获取机器人当前位置的数据，并填入表单中的坐标字段中。在修改界面，用户可以更改原有的点位信息，然后提交更新后的数据，凭借提交的POST请求，后端会更新数据库中的点位信息，提交成功后，系统会显示一个成功消息，并将用户重定向回点位列表页面，保证信息更新即时体现。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-14 点位信息管理-点位修改界面**

**Fig. 5-14 The interface of point modification in point information management**

此外，车位点位关联界面如图4-15所示；定点扫码寻车点位关联界面如图4-16所示；机器人待机位置点位关联界面如图4-17所示。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-15 点位信息管理-车位点位关联界面**

**Fig. 5-16 The interface of parking place and point association in point information management**

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-16点位信息管理-定点扫码寻车点位关联界面**

**Fig. 5-17 The interface of fixed-point QR code car-finding association in point information management**

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-17 点位信息管理-机器人待机位置点位关联界面**

**Fig. 5-18 The interface of robot standby position point association in point information management**

### 机器人管理

机器人信息管理实现了对接入到本系统中机器人的管理，可以对机器人的参数进行配置，包括配置当前运行模式（分为服务模式、暂停服务、调试模式三种模式）。管理人员需要向系统提供连接到服务器的寻车机器人的机器码从而实现绑定。

#### 机器人列表

图5-18展示的是机器人列表界面，其有动态列出系统里所有机器人信息的功能，同时会依据连接状态来显示机器人是否处于在线状态，借助在后台获取机器人的基本信息并结合SocketServer的连接数据，前端得以实时呈现机器人的状态，即在线或者离线。针对每一行数据，管理员可点击执行编辑或者删除操作，删除操作会直接调用API，将数据库中的相应数据删除，并且更新界面。

图形用户界面, 应用程序, 表格, Excel

AI 生成的内容可能不正确。

**图5-18 机器人管理-机器人列表界面**

**Fig. 5-18 The interface of robot list in robot management**

#### 机器人添加

图5-19呈现的是机器人添加界面，管理员可凭借这个页面添加新的机器人，页面设有多个输入字段，像是机器码、机器人名称、描述以及运行模式。运行模式呈现为一个下拉框，管理员可以从中选择服务模式、暂停服务或者调试模式中的一种，提交的时候，表单数据会借助GET请求发送至后端API，后端接收到数据后会把新机器人的信息添加到数据库中，成功之后会跳转回机器人列表页面，并给出操作成功的提示。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-19 机器人管理-机器人添加界面**

**Fig. 5-19 The interface of robot addition in robot management**

#### 机器人修改

图5-20是机器人修改界面，管理员可借助此界面修改已存在的机器人信息。与添加界面相似，修改页面会展示机器人当前的所有信息，并且允许管理员对这些信息进行修改，表单中的机器码、名称、描述以及模式等字段是可以编辑的，修改完成后，管理员提交表单，后端会更新数据库中的机器人信息，修改成功后会返回到机器人列表页面。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-20 机器人管理-机器人修改界面**

**Fig. 5-20 The interface of robot modification in robot management**

### 用户管理及管理平台登录认证功能

用户信息管理功能仅对用户为管理员级别的用户开放（一般员工除本功能外其他功能均可正常使用）。管理员级别的用户可以通过该页面浏览所有的用户信息，可以点击添加用户进行用户信息添加，也可以通过选中一项用户信息进行编辑操作和删除操作。

#### 用户列表

用户列表界面如图5-21所示。用户列表借助GET请求从后端获取所有用户的详细资料，运用Bottle的@app.route('/page/manage\_user\_list')来处理这个请求，后端从数据库取出所有用户的数据，再渲染到模板页面当中，前端页面运用了Bootstrap的表格组件来展示用户信息，包含用户ID、用户名、用户角色以及用户状态。每个用户信息后面设有两个操作按钮：一个是编辑按钮，另一个是删除按钮，删除按钮会调用后端的删除API来删除该用户，编辑按钮则会跳转到修改页面，供管理员去修改该用户的信息。

图片包含 图形用户界面

AI 生成的内容可能不正确。

**图5-21 用户管理-用户列表界面**

**Fig. 5-21 The interface of user list in user management**

#### 用户添加

用户添加界面如图5-22所示。在添加用户界面部分，经由一个GET请求渲染出一个表单，管理员在表单里输入新用户的用户名、密码、角色和状态等信息。前端表单设计采用了Bootstrap样式。管理员填完表单后，依靠提交表单把数据发送到/api/user\_add路径，该路径接收到数据后，先对输入进行验证，接着将新用户的相关信息插入数据库，最后跳转回用户列表页面并给出“添加用户成功”的提示。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-22 用户管理-用户添加界面**

**Fig. 5-22 The interface of user addition in user management**

#### 用户修改

用户修改界面如图5-23所示。修改用户界面通过在用户列表界面中点击编辑按钮之后进行的操作，修改页面依靠GET请求获取到当前用户的信息，并以表单形式呈现出来，管理员可修改该用户的用户名、密码、角色和状态，用户修改完后提交表单，数据借助GET请求发送到后端的/api/user\_edit接口，该接口会更新数据库里的用户数据。修改完成后，后端会返回一个操作成功的提示，管理员会被重定向回用户列表界面。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

**图5-23 用户管理-用户修改界面**

**Fig. 4-23** **The interface of user modification in user management**

#### 登录认证

管理平台登录认证页面如图5-24所示。管理平台的登录认证界面采用了一个登录表单，用户输入用户名和密码后，凭借表单的POST请求将数据发送到后端的/api/login接口进行认证，后端会验证输入的用户名和密码是否正确。

图形用户界面, 网站

AI 生成的内容可能不正确。

**图5-24管理平台登录认证界面**

**Fig. 5-24** **The interface of management platform authentication**

## CPR设备端

CPR设备在本项目中起到了监控车位状态并识别停放车辆的车牌号的作用。CPR设备硬件由树莓派4B（4GB运行内存版本）、摄像头组成，并通过有线网络连接至服务器。CPR设备安装在欲识别车位的前方，支持多个车位的识别匹配。

### 基于YOLOv11和Paddle OCR的车牌-车位识别匹配功能

本算法为寻车机器人系统提供车牌识别服务。该算法的实现步骤为：

首先，服务器定时向停车场内的所有CPR设备发送对应的识别区域信息。识别区域信息包含该CPR设备摄像头视野范围内指定车位所在的图像区域。随后，CPR设备接收到信息之后，通过摄像头采集图像。随后将采集的图像使用训练完成的YOLOv11模型进行车牌识别。识别到的车牌所在位置会与下发的识别区域信息对比，如果车牌位于识别区域内则裁剪车牌图像，并传入Paddle OCR进行文字识别。识别完成之后该算法判断当前车牌号是否符合车牌号格式规范。对于不符合车牌号格式规范或者识别信息为空，则标记当前车位为空闲状态。若当前车位车牌号符合格式，则标记该车位为占用状态，并录入车牌号信息。最终将识别数据发送至服务器。识别流程图可见图5-25。



**图5-25车牌-车位识别匹配功能流程图**

**Fig. 5-25 The flowchart** **license plate - parking space recognition and matching function**

本数据集来自某企业的监控视频，涵盖了多种车牌样式、不同的光照条件和多样化的拍摄角度。数据集包含863张图像，其中包括在白天、夜间、阴天等不同光照条件下采集的车牌图像。此外，数据集中包含多种不同角度的车牌图像，包括正面、侧面以及斜角拍摄的车牌图像。所有图像通过labelImg软件进行人工标注。图5-26为数据集中部分图片



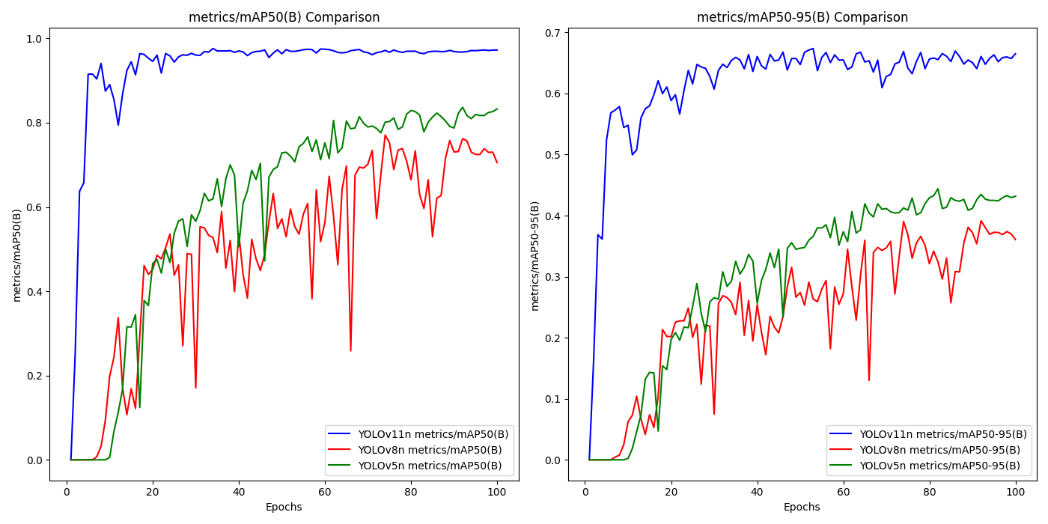
**图5-26 数据集图片**

**Fig. 5-26 The results of** **license plate recognition model validation set recognition**

为了进一步提高车牌检测的精度，本项目选择采用 YOLOv11 模型进行训练。经过 100 轮的训练，最终得到了本项目所使用的高效车牌识别模型。

为了验证本项目所使用的模型就其他模型在车牌识别任务上性能的提高，本项目使用了多个常见的评价指标进行性能提高的判断，包括 mAP、损失函数、精度和召回率、学习率等，通过对不同模型（YOLOv8 和 YOLOv5）在这些指标上的表现进行对比分析，从而深入了解模型的优劣。

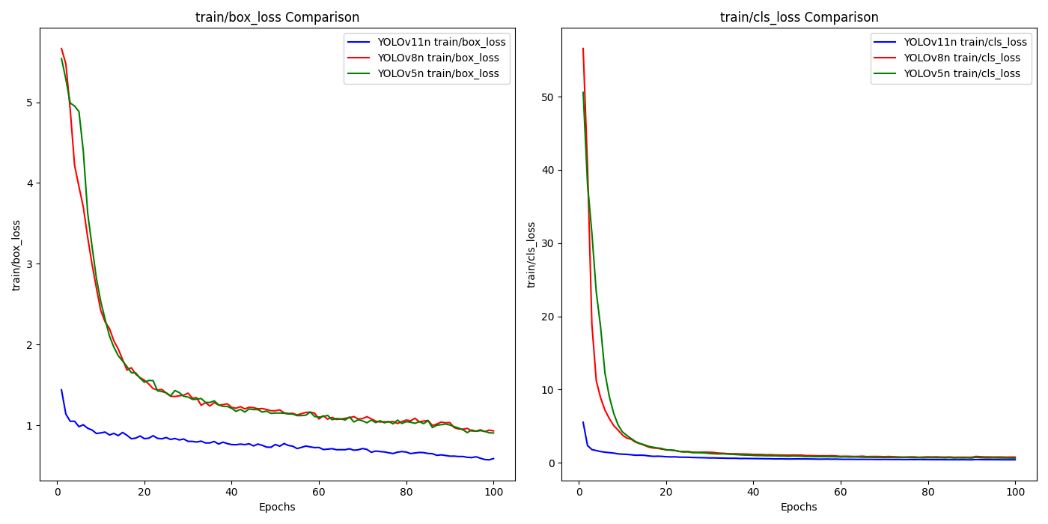
首先，在 mAP（mean Average Precision）方面，YOLOv11 在 mAP50 和 mAP50-95 上的表现显著优于 YOLOv5 和 YOLOv8，如图5-27所示。尤其是在 mAP50-95 这一更为严格的指标上，YOLOv11 的得分明显高于其他模型，这表明 YOLOv11 能在较复杂的车牌识别任务中保持更高的检测精度和鲁棒性。YOLOv5 和 YOLOv8 的表现相对较弱，尤其是在 mAP50-95 上，它们的精度有所下降，未能达到 YOLOv11 的水平，反映出它们在处理不同尺寸和多样化角度的车牌图像时的局限性。



**图5-27 不同YOLO模型在训练过程中的mAP表现对比图**

**Fig. 5-27 The diagram of mAP performance comparison of different YOLO models during training**

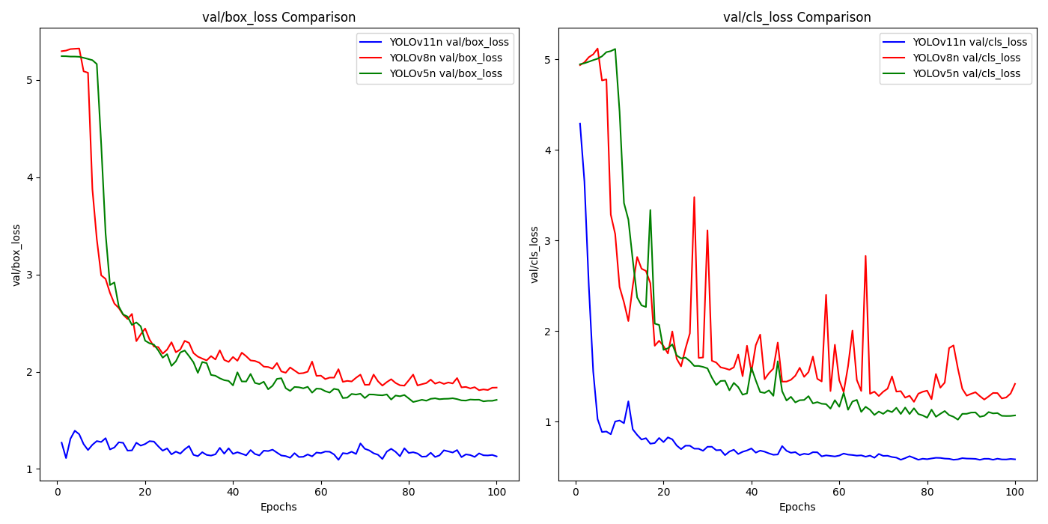
在训练过程中，YOLOv11 展现了较低的训练损失（train/box\_loss 和 train/cls\_loss），这意味着模型在边界框回归和分类任务上的优化效果较好，训练过程稳定且高效，如图5-28所示。与此相比，YOLOv5 和 YOLOv8 在训练损失上稍高，尽管它们仍能完成车牌识别任务，但从损失值来看，YOLOv11 在学习过程中更加精确和优化，尤其是在验证集上的表现。YOLOv11 的验证集损失（val/box\_loss 和 val/cls\_loss）也显著低于其他模型，进一步验证了其在实际应用中的高效性和泛化能力。



**图5-28 不同YOLO模型在训练过程中的训练损失表现对比图**

**Fig. 5-28 The diagram of training loss performance comparison of different YOLO models during training**

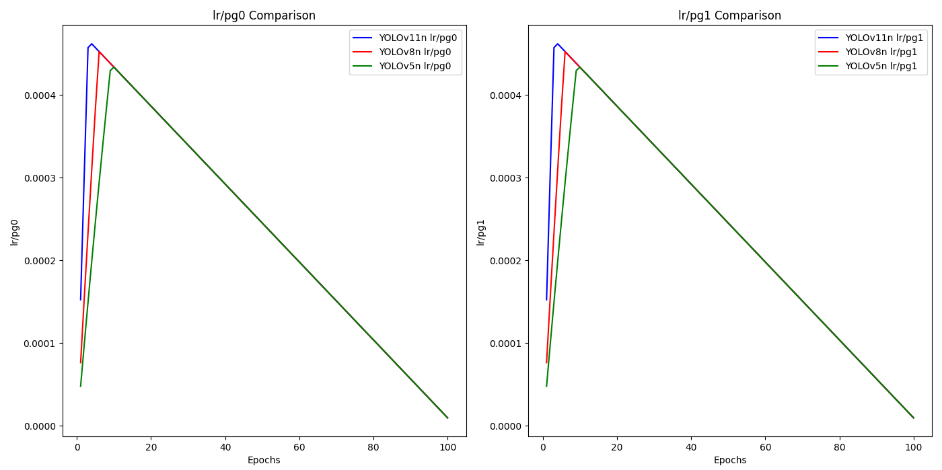
在精度（metrics/precision(B)）和召回率（metrics/recall(B)）方面，YOLOv11 的表现依然突出，如图5-29所示。特别是在精度方面，YOLOv11 能够有效地减少假阳性（False Positives），确保在大多数检测为车牌的图像中，实际为车牌的比例较高。而在召回率方面，YOLOv11 同样表现良好，能够最大限度地捕捉到车牌，减少漏检现象。相较之下，YOLOv5 和 YOLOv8 在精度和召回率之间的平衡表现较好，但在某些复杂或低光照场景下，召回率较低，导致漏检的情况稍显明显。

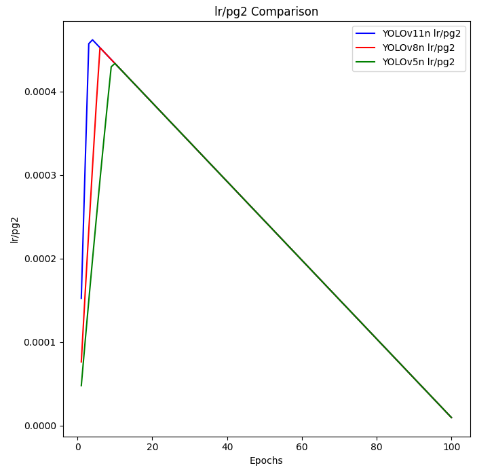


**图5-29 不同YOLO模型在训练过程中的训练精度和召回率表现对比图**

**Fig. 5-29 The diagram of training accuracy and recall performance comparison of different YOLO models during training**

在学习率的调整方面，YOLOv11 展现出了更为灵活的学习率调度策略，如图5-30所示。lr/pg0, lr/pg1, lr/pg2 在训练过程中变化平滑，有效避免了训练过程中的震荡，保证了模型能够平稳地收敛。而 YOLOv5 和 YOLOv8 在学习率的调节上表现得稍有不稳定，可能导致训练过程中的收敛速度较慢，影响了最终的性能表现。





**图5-30 不同YOLO模型在训练过程中的训练学习率的调整表现对比图**

**Fig. 5-30 The diagram of learning rate adjustment performance comparison of different YOLO models during training**

通过对 YOLOv11、YOLOv8 和 YOLOv5 在各个关键指标上的对比，该实验发现 YOLOv11 在车牌识别任务中表现出色，尤其是在 mAP、损失函数、精度、召回率等方面具有明显的优势。虽然 YOLOv5 和 YOLOv8 也表现出了不错的性能，但本项目所使用的 YOLOv11 车牌检测模型在综合能力上占据了明显的优势，适用于更为复杂和多变的车牌识别场景。

## 机器人端

机器人在本项目中起到了反向寻车作用。通过带领车主找到停放的车辆来实现反向寻车。该机器人硬件采用Jetson Nano主控作为上位机，STM32F407VET6芯片主控作为下位机。上位机安装Ubuntu操作系统，搭载ROS框架。下位机负责实现供电和操控总线舵机和电机，同时传输雷达数据。机器人还配置了深度相机，激光雷达，为机器人路径规划、自主导航等任务提供相关环境信息。机器人配有7英寸显示屏，用于实现与用户的视觉交互。图5-31为机器人硬件组装示意图。

图示

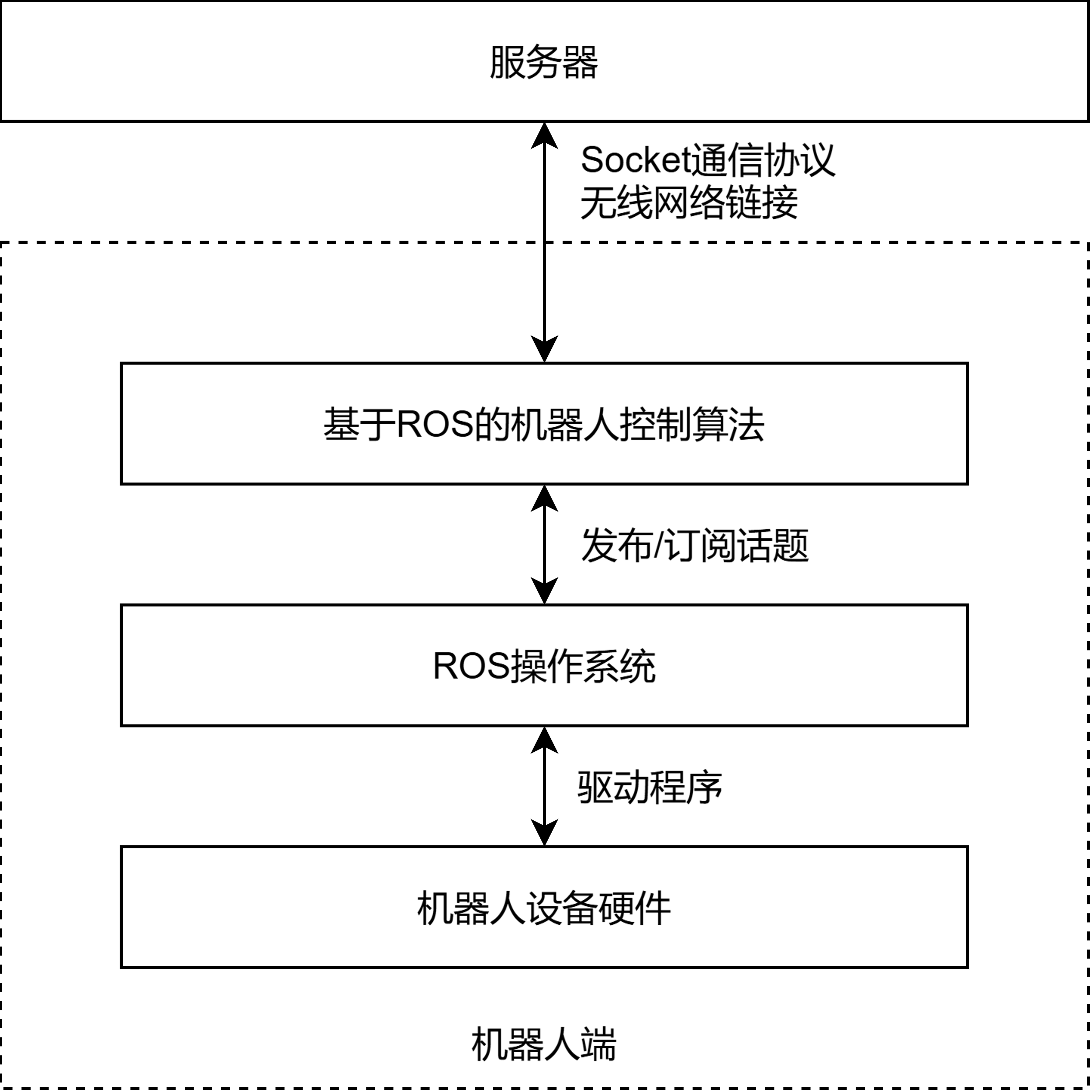
AI 生成的内容可能不正确。

**图5-31 寻车机器人硬件组装示意图**

**Fig. 5-31 The diagram of car searching robot hardware assembly**

### 基于ROS的机器人控制算法

基于ROS的机器人控制算法通过接收服务器的调度数据，操控ROS机器人操作系统来控制机器人的运行移动。该控制算法的层级结构图如图5-32所示。

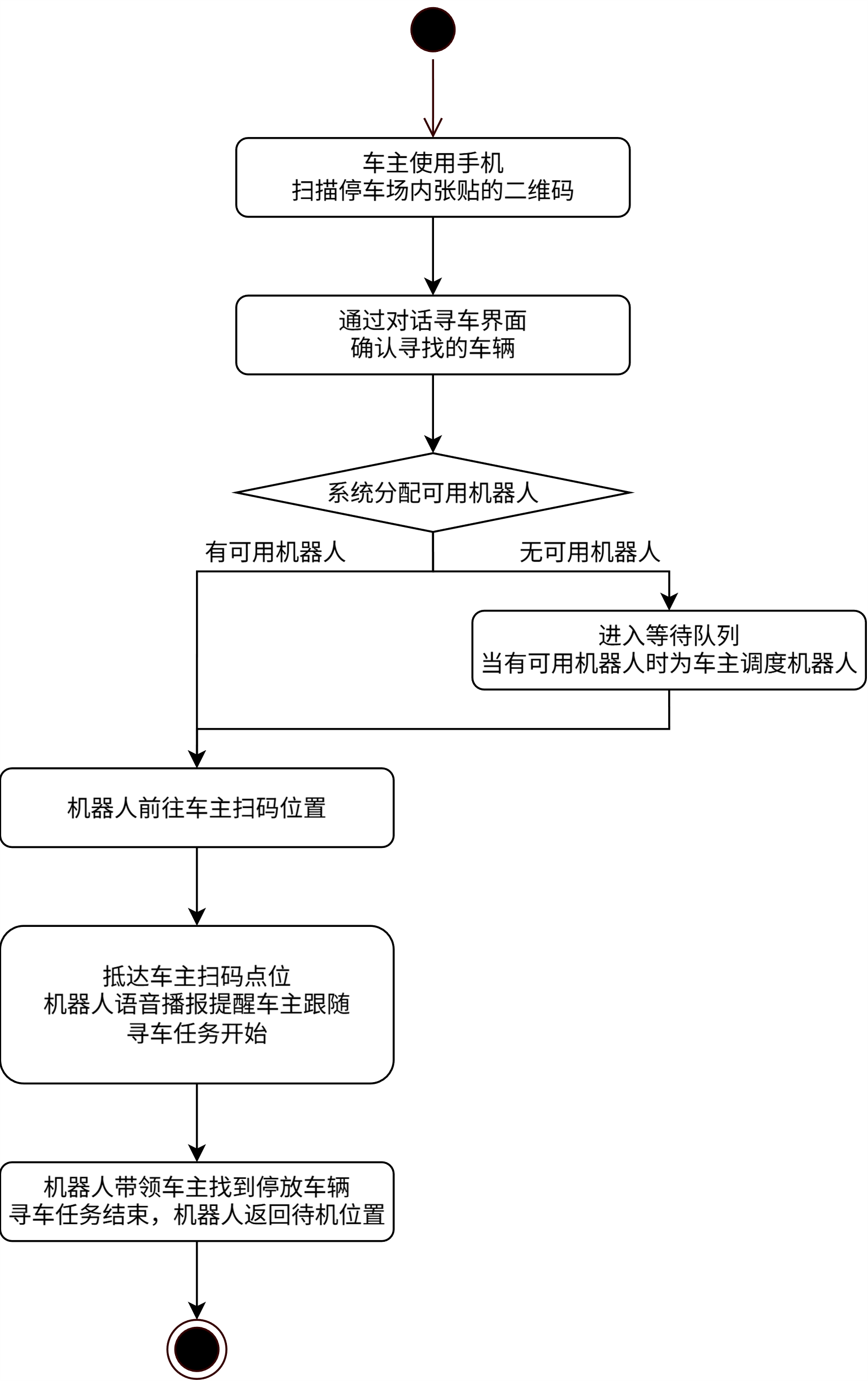


**图5-32 机器人控制算法层次结构图**

**Fig. 5-32 The diagram of robot control algorithm hierarchical structure**

在本系统中，机器人有两种方式带领车主进行寻车，分别是：定点扫码寻车和非定点扫码寻车。

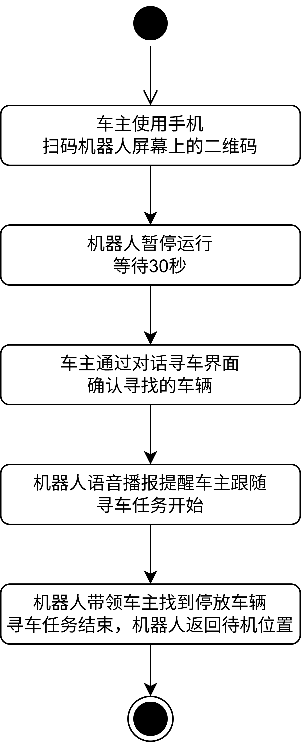
定点扫码寻车需要车主通过手机扫描停车场内张贴的寻车二维码，寻车二维码通过服务器后台预先绑定了一个固定的点位。车主扫描二维码后跳转到前台对话寻车界面，通过和大语言模型对话来确认寻找的车辆，确认寻车后服务器分配距离当前位置距离最近的可用机器人，前往车主扫码位置接应车主。机器人抵达车主所在位置后，停止运行并语音播报提醒车主跟随机器人进行寻车。机器人开始进行寻车任务。当机器人带领车主找到寻找到的车辆后，机器人返回固定待机点位。定点扫码寻车示意图可见图5-33。



**图5-33 定点扫码寻车流程图**

**Fig. 5-33 The flowchart of fixed-point QR code car search**

车主于停车场内行走时，若碰到正返回待机位置的寻车机器人，可扫描机器人屏幕上的寻车二维码以开启非定点扫码寻车流程，车主扫码之后，机器人会停止运行，并等待30秒，该期间，车主需操作手机，借助与大模型对话来确认要寻找的车辆，确认好要找的车辆后，服务器给扫码机器人下达寻车指令，机器人便带领车主前往车辆所在之处。寻车任务结束后，机器人会返回待机点位，非定点扫码寻车示意图如图5-34所示。

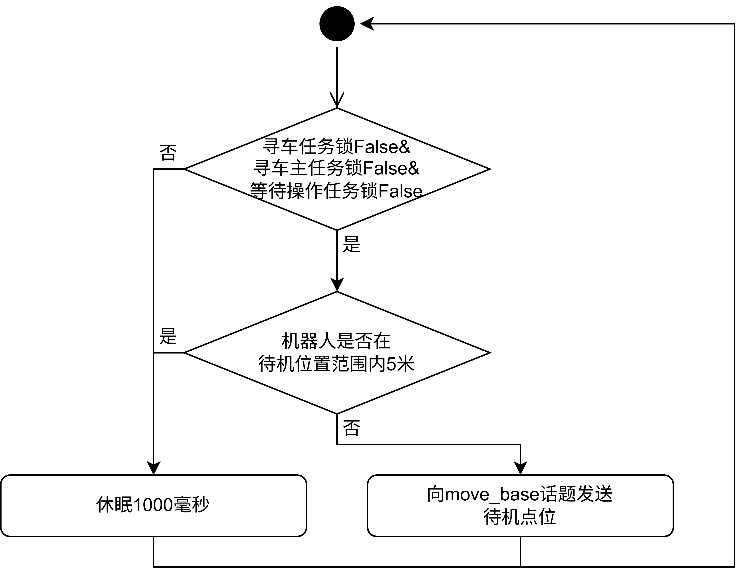


**图5-34 非定点扫码寻车流程图**

**Fig. 5-34 The flowchart of non-fixed-point QR code car-finding**

基于ROS的机器人控制算法基于上述两种寻车方式和ROS机器人操作系统设计。当机器人当前无下发任务时，本算法将检查机器人是否在指定待机点位。若当前机器人不在指定范围内，本算法将控制机器人前往待机点位。如果机器人在待机范围内，本算法将休眠。

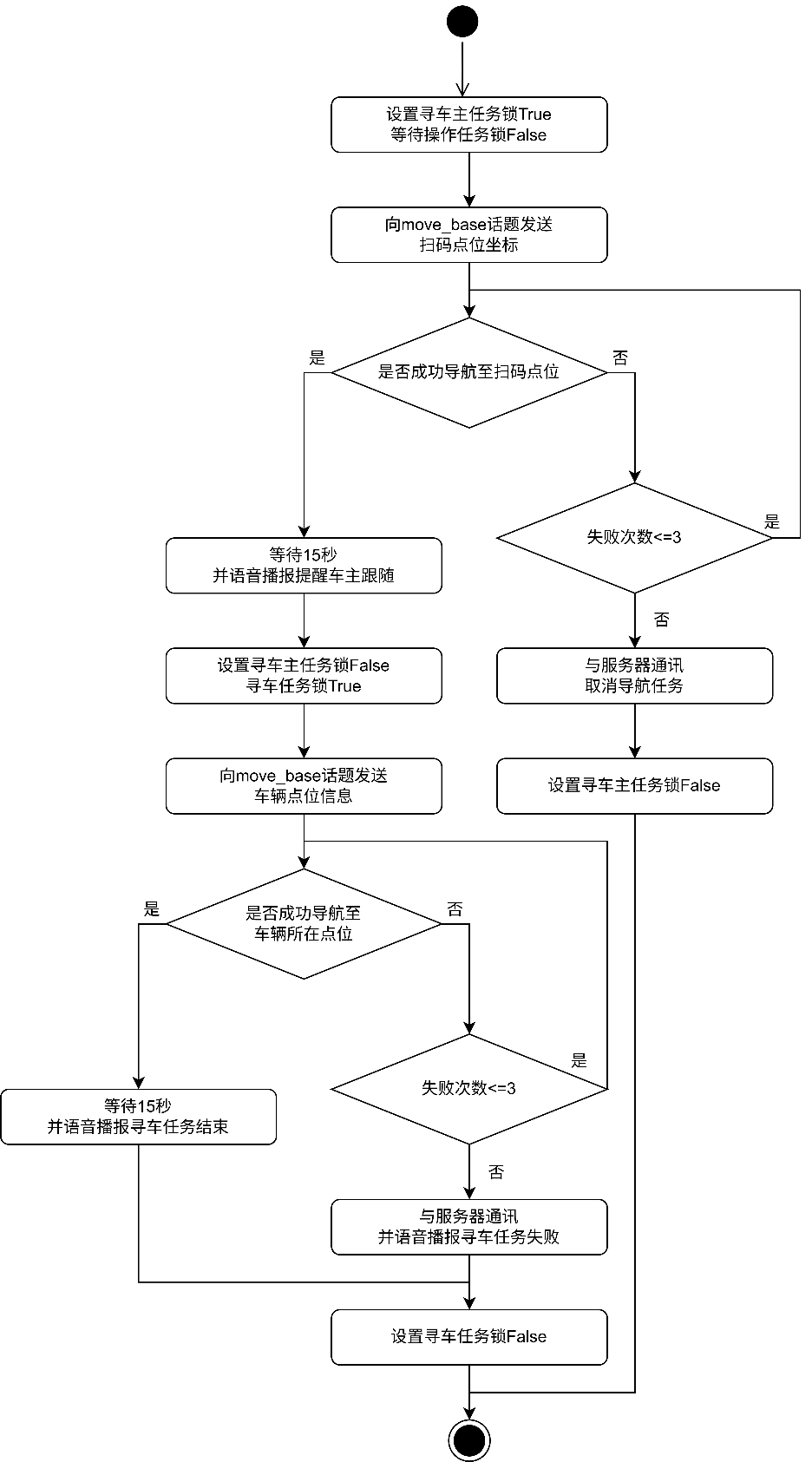
当服务端下发寻车任务时，机器人首先设置锁标志为TRUE，防止机器人执行空闲状态控制算法。随后，机器人按照下发的寻车任务对应的流程运行（定点扫码寻车流程和机器人扫码寻车流程）。运行结束后设置锁标志为FALSE。控制算法流程图如图5-35所示。



**图5-35 机器人空闲状态控制算法流程图**

**Fig. 5-35 The flowchart of robot idle state control algorithm**

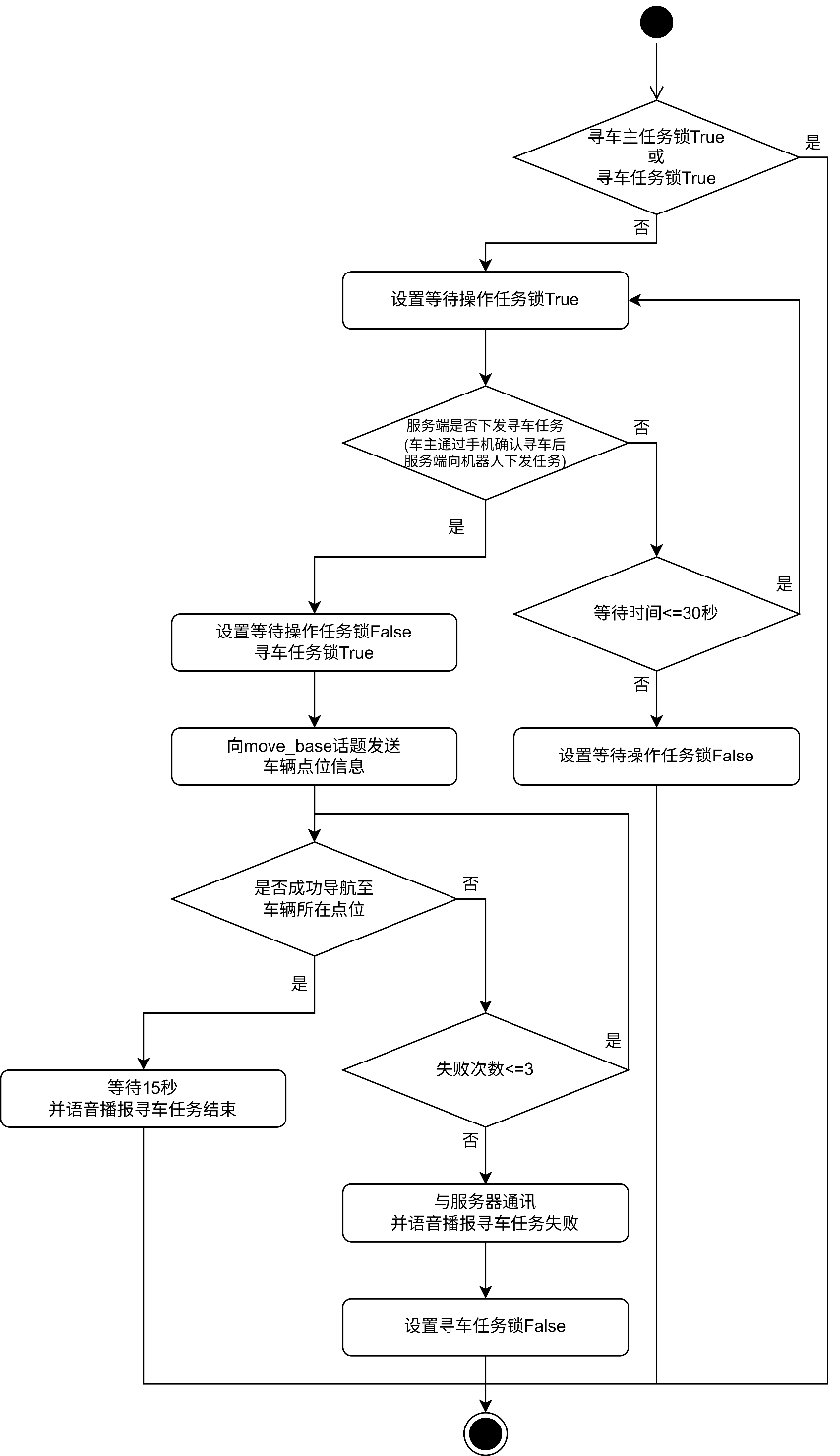
当机器人接收到服务器下发的定点扫码寻车指令后，机器人首先设置寻车主任务锁状态为TRUE，等待操作任务锁状态为FALSE。随后，机器人根据下发指令中的扫码点位向move\_base话题发送扫码点位坐标。在导航过程中，机器人向odom话题通信，检查机器人是否到达扫码点位。如果机器人未成功到达扫码点位并且失败次数大于3次，机器人将向服务器通讯取消导航任务，并设置寻车主任务锁为FALSE。如果机器人成功导航至扫码点位，机器人将等待15秒，并语音播报提醒车主跟随机器人进行寻车。在寻车任务开始前，机器人将设置寻车主任务锁为FALSE，寻车任务锁为TRUE。随后向move\_base话题发送目标车辆所在点位。在到达目标车位前机器人将重复检查当前导航状态。若导航成功，机器人将等待15秒并语音播报导航成功。若导航失败则机器人播报导航失败信息。最后，机器人返回机器人待机点位。定点扫码寻车控制算法流程图如图5-36所示。



**图5-36 定点扫码寻车控制算法流程图**

**Fig. 5-36 The flowchart of control algorithm for fixed-Point QR code car-finding**

当机器人接收到非定点扫码寻车指令时，首先机器人判断当前寻车主任务锁和寻车任务锁状态，以防止正在执行寻车主任务或执行寻车任务的机器人执行新的任务。若当前机器人尚未执行上述任务，机器人将设置等待操作任务锁状态为TRUE。若在30秒内车主未下达寻车任务，则机器人将恢复空闲状态控制算法，并设置等待操作任务锁为FALSE。若车主在30秒内下达寻车任务，机器人将设置等待操作任务锁为FALSE，寻车任务锁为TRUE。并同时向move\_base节点发送目标车位点位信息。在导航至目标车位前，机器人将检查导航状态。若导航成功，机器人将等待15秒并语音播报导航成功。若导航失败则机器人播报导航失败信息。最后，机器人返回机器人待机点位。非定点扫码寻车控制算法流程图如图5-37所示。



**图5-37 非定点扫码寻车控制算法流程图**

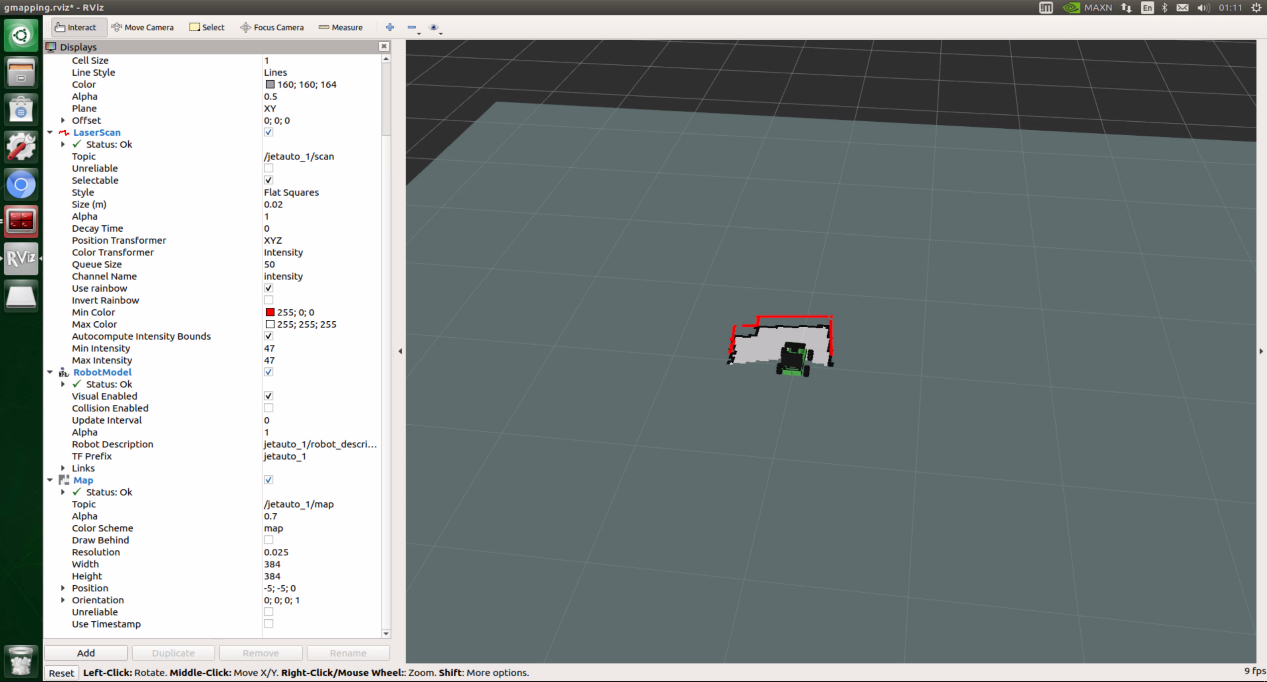
**Fig. 5-37 The flowchart of control algorithm for non-fixed-point QR code car-finding**

在本项目中，SLAM技术是机器人自主导航的核心技术之一。本项目选择使用Gmapping算法进行地图构建，利用navfn全局路径规划插件进行全局路径规划，TEB（Timed Elastic Band）算法进行局部路径规划。

Gmapping是一种基于粒子滤波（RBPF，Rao-Blackwellized Particle Filter）算法的SLAM实现。通过激光雷达等传感器获取环境信息，Gmapping可以在动态环境中构建实时地图，并进行精确的定位。该算法通过多次扫描匹配，结合里程计数据，动态修正机器人的位姿，适应移动过程中的位置变化。其基本的更新过程为：

其中， 表示机器人在时刻 的位置 给定观测和控制信息的后验概率， 是观测模型， 是运动模型， 是机器人在时刻 的控制输入， 是归一化因子。

在机器人进行自主导航时，Gmapping会实时更新地图，以便机器人在未知环境中进行定位和避障。其中，机器人RViz建图界面如图5-38所示。当机器人在停车场或室内环境中移动时，SLAM算法的实时更新能力确保了地图的高精度和低延迟，进而提升了导航的可靠性和效率。



**图5-38 机器人RViz建图界面**

**Fig. 5-38 The interface of robot mapping in RViz**

为了实现机器人从起点到目标点的全局路径规划，本项目使用了navfn算法。navfn是ROS中提供的一个全局路径规划插件，基于Dijkstra算法实现，用于为机器人生成从起点到目标点的最优路径。该算法通过输入地图信息和障碍物数据，计算出一条最短且可行的路径，以便机器人在全局范围内避开障碍物并平稳地前进。

Dijkstra算法的基本公式为：

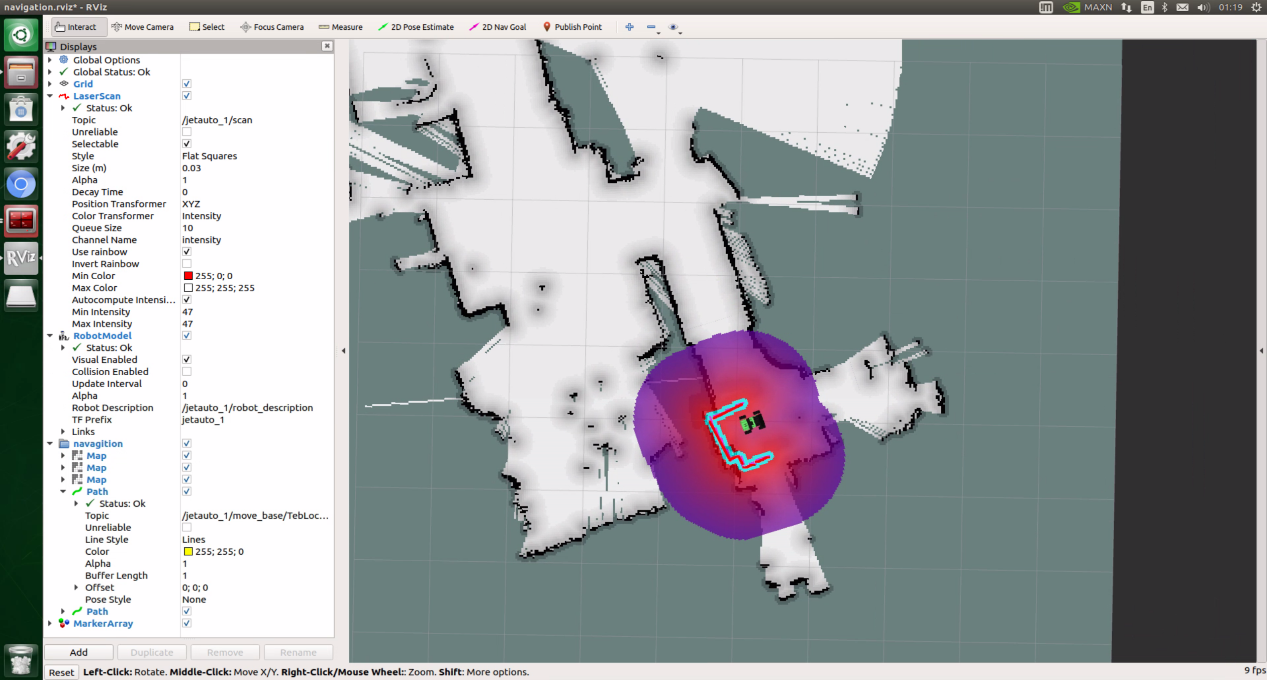
其中， 是从起点到节点 的最短路径， 是与节点 相邻的节点集合， 是节点 到节点 的边权重，表示两节点之间的代价。通过迭代更新，Dijkstra算法能有效地计算出从起点到目标点的最短路径。

navfn的优势在于其高效的路径搜索能力，可以在静态或动态环境中生成准确的导航路径。其核心是基于代价地图的计算，结合全局路径规划的需求，能够在复杂的环境中快速找到最优路线。与局部路径规划相比，navfn主要负责在较大范围内生成初步路径，保证机器人从起始点到目标点的整体导航合理性。

为了确保机器人在复杂环境中的高效避障和精确导航，本项目采用了TEB（Timed Elastic Band）算法作为局部路径规划的核心算法。TEB算法是一种优化路径规划算法，通过考虑时间和空间的约束，动态生成机器人的运动轨迹。TEB的目标是最小化代价函数，通常表达为：

其中， 表示控制输入， 是速度， 是加速度， 是障碍物的距离约束， 是权重系数。TEB算法通过优化这些约束，生成一条平滑的轨迹，确保机器人能在动态环境中顺利通过并接近目标。

与传统的DWA（Dynamic Window Approach）算法相比，TEB算法能够在动态和变化的环境中更好地调整路径，使得机器人在执行任务时，能够平滑地避开障碍物并按照预定轨迹准确行驶。TEB算法的应用使得机器人在接近目标时，能够快速响应并调整运动轨迹，避免了传统路径规划算法在复杂场景中的局限性。其中，机器人RViz导航界面如图5-39所示。



**图5-39 机器人RViz导航界面**

**Fig. 5-39 The interface of robot navigation in RViz**

### 基于Bottle WEBUI的状态显示界面

本界面显示于机器人搭载的触摸屏上。当用户选择机器人扫码寻车时，车主可以通过观察该页面进行扫码，当进行寻车任务时可以从该页面获取机器人当前运行状态。状态显示界面主要由扫码寻车界面、扫码等待页面、寻车页面三部分组成。

机器人在空闲状态时展示扫码寻车页面。扫码寻车页面右方展示一个二维码，车主扫码之后跳转到对话寻车页面，同时机器人的状态显示界面跳转到扫码等待页面，等待30秒。扫码寻车界面如图5-40所示，扫码等待页面如图5-41所示。



**图5-40 扫码寻车界面**

**Fig. 5-40 The interface of QR code car search**



**图5-41 扫码等待界面**

**Fig. 5-41 The interface of QR code scanning waiting**

当用户确认寻找的车辆，服务器下发任务至机器人后，状态显示界面将显示寻车界面。寻车界面显示寻找的车位的编号和目标车辆车牌号，如图5-42所示。当寻车任务结束后，状态显示界面将提示车主寻车任务结束，并返回扫码寻车界面。



**图5-42 寻车界面**

**Fig. 5-42 The interface of car searching**

# 系统测试

## 测试目的

为验证本系统各项功能均能正常稳定运行，确保数据传输的准确性，以及机器人控制、车牌识别等核心功能的可靠性。因此需对本系统进行详细的系统测试。

## 功能测试

**表6-1 车牌识别模块功能测试记录表**

**Tab. 6-1 The table of license plate recognition module function test**

|  |  |  |
| --- | --- | --- |
| 测试用例 | 预期结果 | 实际结果 |
| 在不同光照、角度和车牌样式下进行车牌识别测试 | 车牌号应被正确识别，并能传输至数据库 | 车牌号识别准确，符合预期，数据传输无误 |

**表6-2 寻车模式模块功能测试记录表**

**Tab. 6-2 The table of car finding mode module function test**

|  |  |  |
| --- | --- | --- |
| 测试用例 | 预期结果 | 实际结果 |
| 测试定点扫码寻车模式，车主扫码后，机器人能准确导航至车位 | 车主扫码后，机器人能够精确到达车主指定位置 | 定点扫码寻车模式顺畅，导航准确 |
| 测试机器人扫码寻车模式，车主扫码后，机器人能准确带领车主找到车辆 | 扫码后，机器人开始任务并成功引导车主找到车辆 | 机器人扫码寻车功能正常，任务顺利完成 |

**表6-3用户界面模块功能测试记录表**

**Tab. 6-3 The table of user interface module function test**

|  |  |  |
| --- | --- | --- |
| 测试用例 | 预期结果 | 实际结果 |
| 测试车主与系统的交互界面，包括文本与语音输入方式 | 用户能方便地查询车辆，并顺利开始寻车任务 | 界面操作简单，语音与文本输入都能顺畅进行 |
| 测试车主在界面上找到寻车二维码，并能正确扫码操作 | 用户能顺利完成扫码操作并进入寻车任务页面 | 扫码界面加载迅速，用户反馈良好 |

**表6-4 数据库同步模块功能测试记录表**

**Tab. 6-4 The table of database synchronization module function test**

|  |  |  |
| --- | --- | --- |
| 测试用例 | 预期结果 | 实际结果 |
| 测试CPR设备和服务器之间的数据同步功能 | 车位信息能够实时更新，车牌识别信息无误传输至数据库 | 数据同步及时，车位信息与停车场状态一致 |
| 测试数据在高并发情况下的同步能力 | 即使在高并发情况下，数据同步依然准确无误 | 高并发情况下，数据同步无明显延迟 |

**表6-5 机器人控制模块功能测试记录表**

**Tab. 6-5 The table of robot control module function test**

|  |  |  |
| --- | --- | --- |
| 测试用例 | 预期结果 | 实际结果 |
| 测试基于ROS的机器人控制，检查机器人是否能准确执行寻车任务 | 机器人能根据任务指令，准确完成寻车并返回待机点 | 机器人响应迅速，能够按指令精确执行任务 |
| 测试机器人在任务完成后的待机状态 | 任务完成后，机器人能够准确进入待机状态，准备接受下一指令 | 机器人能顺利进入待机状态，无异常 |

**表6-6 用户信息管理模块功能测试记录表**

**Tab. 6-6 The table of user information management module function test**

|  |  |  |
| --- | --- | --- |
| 测试用例 | 预期结果 | 实际结果 |
| 测试管理员是否能够成功添加、修改和删除用户信息 | 管理员能够顺利添加新用户，修改和删除已有用户信息 | 用户管理功能正常，能够正确进行增、删、改操作 |
| 测试用户登录认证功能，管理员是否能够通过正确用户名和密码登录后台系统 | 输入正确的用户名和密码，管理员成功登录后台 | 登录认证功能正常，管理员能成功登录后台管理系统 |

**表6-7 CPR设备管理模块功能测试记录表**

**Tab. 6-7 The table of CPR device management module function test**

|  |  |  |
| --- | --- | --- |
| 测试用例 | 预期结果 | 实际结果 |
| 测试管理员是否能顺利添加、修改或删除CPR设备信息 | 管理员能够成功添加新的CPR设备，并修改或删除设备信息 | CPR设备管理功能正常，设备信息能够成功添加、修改、删除 |
| 测试CPR设备状态更新功能，管理员能否查看设备的实时状态 | 系统能够实时显示设备的最新状态，并反映出设备的运行情况 | 设备状态同步无误，管理页面能实时显示设备状态 |

**表6-8机器人信息管理模块功能测试记录表**

**Tab. 6-8 The table of robot information management module function test**

|  |  |  |
| --- | --- | --- |
| 测试用例 | 预期结果 | 实际结果 |
| 测试管理员能否正确查看、添加、修改或删除机器人信息 | 管理员可以成功查看、修改机器人信息，并删除不需要的机器人记录 | 机器人信息管理功能正常，管理员可以顺利操作 |
| 测试管理员是否能对机器人的服务模式进行修改（如暂停服务、切换模式等） | 管理员可以修改机器人的工作模式并生效 | 模式切换成功，机器人能够根据新的设置工作 |

**表6-9点位信息管理模块功能测试记录表**

**Tab. 6-9 The table of point information management module function test**

|  |  |  |
| --- | --- | --- |
| 测试用例 | 预期结果 | 实际结果 |
| 测试管理员是否能正确查看、添加、修改或删除点位信息 | 管理员能够成功查看所有点位，并能添加、修改、删除点位 | 点位管理功能正常，点位数据准确 |
| 测试管理员通过机器人回传坐标功能是否能够精确添加点位 | 机器人在调试模式下，管理员能够通过回传坐标准确添加点位 | 点位坐标回传功能准确，操作流程顺畅 |

**表6-10 登录认证模块功能测试记录表**

**Tab. 6-10 The table of login authentication module function test**

|  |  |  |
| --- | --- | --- |
| 测试用例 | 预期结果 | 实际结果 |
| 测试后台管理系统的登录功能，管理员是否能够凭借正确的用户名和密码登录 | 管理员输入正确的用户名和密码能够顺利登录后台管理系统 | 登录认证功能正常，管理员能够正常登录 |
| 测试系统是否能正确限制非法用户登录（例如错误密码或用户名） | 输入错误的用户名或密码时，系统应拒绝登录 | 非法登录被拒绝，系统安全性强 |

## 测试结论

通过各模块的功能测试，智能寻车系统的核心功能如车牌识别、寻车模式、用户界面、数据同步和机器人控制均能够正常稳定运行。整体而言，系统已达到设计预期，并能在实际应用中提供可靠的智能寻车服务。

# 总结与展望

## 本文工作总结

本文设计并成功实现了一种基于 SLAM的智能寻车机器人系统，此系统可缓解停车场中车主寻找停车位时所面临的困境，凭借对停车场环境复杂性以及其给机器人导航系统带来的挑战展开分析，一种集成多种技术的解决办法，该系统依靠 SLAM 技术、YOLOv11 车牌识别算法、Paddle OCR 文字识别技术，以及 ROS 框架支持下的机器人控制，成功达成了机器人在动态变化的停车场环境里的自主导航与定位。

系统的设计主要分成服务器端、CPR端和机器人端这三部分，服务器端负责处理数据以及进行任务调度，CPR端开展车牌识别以及停车位匹配工作，机器人端依据服务器的指令去执行寻车任务，系统的开发运用 Python、MySQL 和 Bottle 框架，保证了数据处理有高效性和实时性。经过实验测试，验证了系统的核心功能，像车牌识别、路径规划、机器人控制等都可稳定运行，契合实际应用的需求。

## 存在不足及未来展望

虽然本文所提出的停车场智能寻车机器人系统在不少方面收获了较为理想的成效，然而仍存在一些技术瓶颈与不足，系统的车牌识别算法在复杂光照以及不同角度下的准确性以及提升空间，尽管YOLOv11和Paddle OCR在多数情形下能提供较精准的识别，但是在特殊环境中可能会有误识别的状况发生。机器人面对动态障碍物时的避障能力有待提高，在停车场人员密集或者出现其他突发状况时，机器人可能会因传感器精度和算法的限制，出现导航偏差或者碰撞风险，

未来研究可从以下几方面开展：其一，优化车牌识别算法，结合更多种类传感器信息，提高复杂环境下的识别率，其二，加强机器人避障功能，借助引入更精确的传感器和实时反馈机制，提升机器人在动态环境中的导航能力，其三，探索智能停车系统与城市交通管理系统的深度融合，实现大规模停车管理的自动化与信息化，推动智慧城市发展。

# 参考文献

1. 李月, 王善勤, 肖宏飞. 基于物联网的智能停车系统设计[J]. 滁州职业技术学院学报, 2024, 23(04): 58-62.
2. Ditta, A., Ahmed, M. M., Mazhar, T., Shahzad, T., Alahmed, Y., & Hamam, H. (2025). Number plate recognition smart parking management system using IoT. Measurement: Sensors, 37.
3. 幸敏, 冼锂东, 黄楚婷, 莫富奇. 基于ZigBee组网的停车场管理系统设计[J]. 物联网技术, 2024, 14(05): 140-143.
4. 陈黎明, 张帅, 吴梦帆. 轻量便捷化的停车场反向寻车系统设计[J]. 电子产品世界, 2024, 31(01): 16-19+28.
5. Rani, R., Kumar, S., Pippal, S. K., Gund, M., Chaudhari, U., Agrawal, R., Dalsaniya, M., & Verma, L. (2024). IPS: Intelligent Parking System Using YOLO and Image Processing. International Journal on Transport Development and Integration, 8(3).
6. 麻吉辉, 王丽杰, 赵原真, 孙建波. 智能停车场反向寻车系统设计[J]. 哈尔滨理工大学学报, 2023, 28(04): 23-32.
7. Ozkaya, M., & Turunc, A. (2025). A Reference Architecture for Smart Car Parking Management Systems. Systems, 13(2).
8. Sahu, D., Sinha, P., Prakash, S., Yang, T., Rathore, R. S., & Wang, L. (2025). A multi-objective optimization framework for smart parking using digital twin Pareto front MDP and PSO for smart cities. Scientific Reports, 15(1).
9. 周伟. 多AGV系统在智能停车场中的调度研究[D]. 导师：秦实宏; 吴敏. 武汉工程大学, 2023.
10. 李今, 孙颖, 李赞赞, 刘明涛. 基于物联网技术的智能停车场系统设计[J]. 电子质量, 2024, (05): 6-11.
11. Hudda, S., Barnwal, R., Khurana, A., & Haribabu, K. (2024). A WSN and vision based smart, energy efficient, scalable, and reliable parking surveillance system with optical verification at edge for resource constrained IoT devices. Internet of Things, 28.
12. 孔德财, 崔杰, 汤怡, 刘瀚辰. 智能停车系统研究综述[J]. 物流工程与管理, 2022, 44(09): 109-111+108.
13. 方利国, 方曦. Python实现教程[M]. 化学工业出版社, 2024.
14. 殷锡亮, 刘阳, 张胜扬. Python程序设计[M]. 人民邮电出版社, 2023.
15. 马兴录, 王剑峰, 刘扬. 人工智能应用开发[M]. 化学工业出版社, 2023.
16. 李月军. 数据库原理与MySQL应用[M]. 人民邮电出版社, 2022.
17. 余斌. 基于Bottle的Python网络应用开发[J]. 无线互联科技, 2014, (06): 29+103.
18. 胡春旭. ROS机器人开发实践[M]. 机械工业出版社, 2023.
19. 李忠新, 余鹏飞, 梁振虎. 机器人SLAM导航技术与实践[M]. 化学工业出版社, 2023.