

河北科技大学

毕 业 设 计

学生姓名： 李圭印 学 号： 170707109

学 院： 信息科学与工程学院

专 业： 物联网工程

题 目： 基于嵌入式硬件的高压巡线

机器人的开发与设计

指导教师： 许云峰(副教授)

评阅教师： _____

2021 年 5 月

河北科技大学本科生 毕业设计原创性声明

毕业设计题目：基于嵌入式硬件的高压巡线机器人的开发与设计

学生姓名：李圭印

专业班级：物联网 171

学号：170707109

本人郑重声明：所呈交的毕业设计是本人在指导教师的指导下，独立进行研究工作所取得的成果。内容中除特别注明和引用外，均为本人观点，不存在剽窃、抄袭他人的学术观点、思想和成果，不存在伪造、篡改实验数据。如有违规行为发生我愿承担一切责任，接受学校的处理，并承担相应的法律责任。本人完全意识到本声明的法律结果由本人承担。

学生（签名）：李圭印

2021 年 5 月 18 日

河北科技大学毕业设计成绩评定表

姓 名	李圭印	学 号	170707109	成 绩	95
专 业	物联网工程				
题 目	基于嵌入式硬件的高压巡线机器人的开发与设计				
指导教师评语及成绩	<div>成绩：95</div> <div>按期圆满完成任务书规定的任务；能熟练地综合运用所学理论和专业知识，立论正确，程序设计正确、严密，合理。</div> <div>独立工作能力较强，科学作风严谨；设计有自己独到见解，水平较高。</div> <div>指导教师：_____年____月____日</div>				
评阅教师评语及成绩	<div>成绩：96</div> <div>说明书条理清楚，论述充分，文字通顺，符合技术用语要求，符号统一，编号齐全，书写工整。</div> <div>评阅教师：_____年____月____日</div>				
答辩小组评语及成绩	<div>成绩：95</div> <div>答辩时，思路清晰，论点正确，回答问题有理论根据，基本要领清楚，对主要问题回答正确、深入。</div> <div>答辩小组组长：_____年____月____日</div>				
答辩委员会意见	<div>同意</div> <div>答辩委员会主任：_____年____月____日</div>				

毕 业 设 计 中 文 摘 要

智能机器人替代电工在高压电缆上进行故障巡检是当前特种机器人领域的重要研究方向，可全面提高高压线路的自动化管理水平，降低发生安全事故的风险。高压线巡检机器人的基本设计 requirements 是能够平稳行驶，安全越障，核心目标是检测高压线以及杆塔上的电工、攀爬人员、鸟巢和缠绕物。现有的高压线路巡检机器人系统主要应用于地线巡检，应用范围广泛，运行稳定。

本文对高压线巡检机器人的结构设计进行分析总结，指出各个阶段机器人模型的优缺点，并最终采用地线巡检方式。之后根据高压巡检的任务需求和精度要求，分析当前研究领域所采用的相关研究技术，从而确定最终方案，即以树莓派为控制平台，百度飞桨为深度学习框架，YOLO V3 为目标检测算法，进而满足高压巡线各个方面的要求。

关键词 巡检机器人 树莓派 目标检测 百度飞桨 YOLO

毕 业 设 计 外 文 摘 要

Title Development and design of high voltage line
patrol robot based on embedded hardware

Abstract

Intelligent robot instead of electrician in high voltage cable inspection is the important research direction in the field of special robot, can comprehensively improve the level of the automation management for high voltage line, and reduce the risk of safety accidents. The basic design requirement of the high-voltage line inspection robot is to be able to run smoothly and cross the obstacles safely. The core goal is to detect electricians, climbers, bird nests and entanglements on high-voltage lines and towers. The existing high-voltage line inspection robot system is mainly used in ground line inspection, which has a wide range of applications, and stable operation.

In this paper, the structural design of the high-voltage line inspection robot is analyzed and summarized, and the advantages and disadvantages of each stage of the robot model are pointed out. Finally, the ground line inspection method is adopted. After that, according to the task requirements and precision requirements of high-voltage patrol, relevant research technologies used in the current research field are analyzed to determine the final scheme, which takes Raspberry Pi as the control platform, PaddlePaddle as the deep learning framework, and YOLO V3 as the target detection algorithm, so as to meet the requirements of various aspects of high-voltage patrol.

Key Words Patrol Robot Raspberry Pi PaddlePaddle YOLO

目 录

1 绪论	1
1.1 研究背景及意义	1
1.2 研究现状	1
1.3 研究内容	4
2 相关技术概述	8
2.1 搭载平台	8
2.2 Raspberry Pi 4B	13
2.3 Raspberry Pi 4 camera	14
2.4 深度学习	16
2.5 目标检测-YOLOv3	21
2.6 本章小结	23
3 高压线巡检机器人系统设计	24
3.1 机械系统设计	24
3.2 控制系统设计	25
3.3 本章小结	32
4 高压线异常检测研究	33
4.1 实验环境	33
4.2 模型训练	33
4.3 模型部署	39
4.4 模型预测	40
4.5 关键技术	40
4.6 本章小结	41
5 实验过程	42
5.1 机器人实现	42

5.2 模型制作.....	43
5.3 移动端预测.....	44
5.4 本章小结.....	47
结 论.....	48
参 考 文 献.....	50
致 谢.....	52
附录 A.....	54
附录 B.....	55

1 绪论

1.1 研究背景及意义

目前我国使用的高压电力输电线的专用线路通常直接建设在一些地形复杂,蜿蜒弯曲的平原地区,同时环境条件也时常极端恶劣。以现代高压电力输电系统线路的高压巡线管理机器人技术作为主要技术研究应用对象,将现代智能高压巡线管理机器人广泛应用于我国现代高压电力系统的各个领域,改善了我国高压电力输电系统线路在现代智能化高压巡线中的技术水平,从而进一步有效确保了高压电力系统安全稳定顺畅运行^[1]。机器人巡检可以实现对重点路段、人工难以攀爬线路进行远程监视。而这些目标的实现均离不开具有自主学习、智能远程控制、能够处理多种任务的智能巡线机器人,通过采用先进的巡检方式工具和设备、可以极大地提高巡检效率,缩短线路故障处理时间,快速恢复用电,消除潜在的隐患,从而保证电力系统的安全运行^[2]。另外,巡检机器人符合科学研究种的人文精神,并将有力地促进电磁屏蔽,环境定位等技术的探索及其他特种作业机器人的开发与设计^[3]。

1.2 研究现状

目前国内外电力机构开展的架空输电线路巡检主要采用人工方法,又可细分为人工上线巡检和在地面使用望远镜检查两类^[3],依靠电力从业人员丰富的经验,这两种方式均具有较高的检测准确率,但前者需要沿线在高空行走,风险性高,后者则需要面对野外环境行走不便,视野中干扰因素多等问题。

近年来,随着科技与经济的进步,大多数电力公司使用无人机或直升机进行巡检,直升机巡检时需与线路保持一定安全距离,降低了检测精度,而无人机巡检不仅需保持安全距离并且续航时间短^[2]。并且这两种方法均受天气影响大,恶劣天气下无法巡检,受电磁干扰大,并且效费比差,成本高昂^[1]。

相比于以上两种巡线方式,机器人地面巡检有效地避免了上述弊端,可以实现对输电线路的常规检测和预警。巡线机器人集合了机械设计、电源技术、通信技术、自动控制技术和传感器技术,能够按预定程序设计实现在复杂环境下对架空输电线路的实时监测,因此,机器人巡检可用于常规高压线巡检与故障简易维修,具有准确判别故障的特点。

1.2.1 机器人模型现状

当前电网的巡线方式主要有地面目测法、航测法和机器人巡检法三种^[2]，但前两种方法具有很大弊端。地面目测法依赖于携带检测设备的巡检人员进行巡检，任务繁重，检测精度低，而且容易发生安全事故；航测法主要通过直升机和无人机两种机型进行巡检^[4]，虽然巡检速度快，但需与电线保持一定的安全距离，况且受天气和电磁干扰影响，成本高昂；机器人巡检法又分为地面行走机械手和线路行走机器人，该类型机器人可有效解决以上两种巡检方式弊端。

1.2.2 深度学习平台现状

随着对于大数据、计算力和现代数字化信息技术体系进行不断优化日臻成熟，深度学习机器学习信息技术在现代计算机图形视觉、语音图像识别、自然语言处理等现代科学信息技术应用领域屡次应用取得最优甚至远远超过超越人类的效果。为了让 AI 专业开发者更高效地实现深度学习算法，各大 AI 科技公司纷纷将深度学习中常用的算子、基础网络结构、迭代优化算法，以及经典的模型实现采用开源的方式形成深度学习框架^[5]。业界知名的深度学习框架包括谷歌的 TensorFlow，Facebook 的 PyTorch 和百度的飞桨（PaddlePaddle）等^[6]。

谷歌于 2018 年 1 月发布 Cloud AutoML 平台，是业内较早发布的一站式机器学习开发平台。发布之初支持定制化图像分类功能，目前已经支持定制化视觉、自然语言处理和结构化数据三大方向。该平台的主要特性是由谷歌的 AutoML 和迁移学习技术驱动，从数据到服务的一站式体验和集成众包模式的数据标注服务。

百度于 2017 年 9 月上线了百度飞桨 EasyDL 平台，是这些主流平台中最早发布和上线可用的一站式深度学习定制开发平台。目前，EasyDL 已支持图像分类、物体检测、图像分割、声音分类、文本分类和视频分类等六大定制方向。该平台的主要特性是基于百度飞桨深度学习框架，支持飞桨 Master 模式的预训练模型，并采用百度 AutoDL 技术实现了只需少量数据就能训练出高精度模型的效果。

1.2.3 目标检测现状

近些年来，随着我国深度学习技术的迅速进步和发展，在各个技术领域当中的研究和应用也逐步成熟，自从 2014 年 Girshick R 等在目标检测领域成功应用深度神经网络后，目标检测的研究方向就基本上被深度学习相关的算法和框架所占领。此外，从整体算法框架和处理思路上看，深度学习在检测问题上的算法大致分为两种^[7]：一种是基于候选区

域生成的检测算法,此算法利用深度神经网络对区域特征进行提取以及最后进行分类,这类方法以 RCNN、Fast-RCNN、Faster-RCNN 系列算法为代表;另一种是基于回归的检测算法,即通过深度学习框架直接回归出图像中的目标对象以及分类,主要以 YOLO、SSD 等算法为代表。

目前,输电线路巡检关键部件检测主要分为传统图像处理法和深度学习法^[8]。Zhai 等提出了一种基于空间形态特征的玻璃和陶瓷绝缘子串故障检测算法,并将其集成到无人机检测系统中;Sampedro 等实现了一种有监督学习的方法来解决杆塔检测和分类问题,其中使用方向梯度直方图特征训练两个多层神经网络,用于解决电力线路检测中的杆塔检测与分类问题;师飘提出了两种基于直方图特征的鸟巢检测算法,并利用主成分分析等方法对其进行聚类分析;焦红完成了改进霍夫变换的防震锤识别定位研究。然而,传统的图像识别检测技术检测种类单一,受外界环境影响波动大,难以投入到实际应用中。

1.2.4 控制器现状

目前常见的控制器有 Arduino、STM32 单片机和树莓派。他们都属于 ARM 系列的处理器,用来实现算法的功能。从内核上讲树莓派,一般是 4 个 ARM 内核,STM32 大部分是 1 个 Cortex-M 系列的内核,Arduino 大部分是 ATMEGA 的 AVR RISC 内核,由此可见树莓派处理性能比较强大;从软件层面讲,树莓派一般使用 OS,几乎所有的应用都是在系统之上进行开发,很少进行裸机开发,STM32 可以移植 RTOS,也可以进行裸机开发,Arduino 因为 IDE 的限制基本上只能进行裸机开发。针对本研究课题,操作系统的环境是 Linux,且要部署 PaddleLite,依次来实现 YOLO V3 的目标检测,第一是操作系统的要求,第二是 CPU 和 GPU 的处理性能,第三是此项目的开发难度,基于以上三点,树莓派更适合作为本课题的控制平台。

树莓派又有多种型号,主流型号为 3B、3B+、4B rev1.1、4B rev1.2、4B rev1.4(其中 rev 为 PCB 版本号)。4B 是目前的主流型号,无论从 CPU 运行速度,文件读取速度、线程调度能力、USB 数据传输等方面,都是碾压 3B、3B+,因此选用 4B,同时 rev1.1 的电路有较大问题,例如 USB 供电电路和 WIFI 电路,而 1.2 和 1.4 均对这两个电路问题进行的改进,且 1.4 更加稳定,但 1.2 并不影响使用,并且考虑到树莓派的性价比,故选择了 rev1.2,另外 4B 模块共有 RAM2G、4G、8G 可选,2G 主要用来搭建比较小型的树莓派控制工程,4G 主要用来运行深度学习模型,而 8G 适合将树莓派作为远程服

务器使用，因此 4G 更加适合本课题，所以最终选择了 Raspberry Pi 4B 4G rev1.2 型号的树莓派。最后还要对树莓派系统进行选型，考虑到树莓派本身 ARM 为 64 位架构，且要运行 Paddle Lite 深度学习模型，对操作系统的实时吞吐量和计算量有较高的要求，故选择 64 位系统，且 Linux 系统更适合开发产品，其可操作性远远大于 Windows。另外 Ubuntu 系统和树莓派的兼容性较差，故选择了 Debian 系统，综合以上，安装了树莓派官方的 raspbian-buster-arm64 位系统。

1.3 研究内容

本课题研究项目为基于嵌入式的高压巡线机器人的开发与设计，主要解决高压输电线异常状态以及异物监测的问题。本研究的主要目的是设计搭建一款能用于高压输电线巡检的地面机器人平台，基于该平台进行移动机器人定位建图、深度学习及目标检测方法的研究。项目研究的机器人系统由机械本体、树莓派 4B、上位机以及各类传感与执行元件组成。传感设备包含深度相机，单线激光雷达，超声波及摄像头等，用以实现高压电缆关键部件识别、自动行驶与避障，线缆损伤检测及越障过程中的定位等功能。具体工作内容如下：

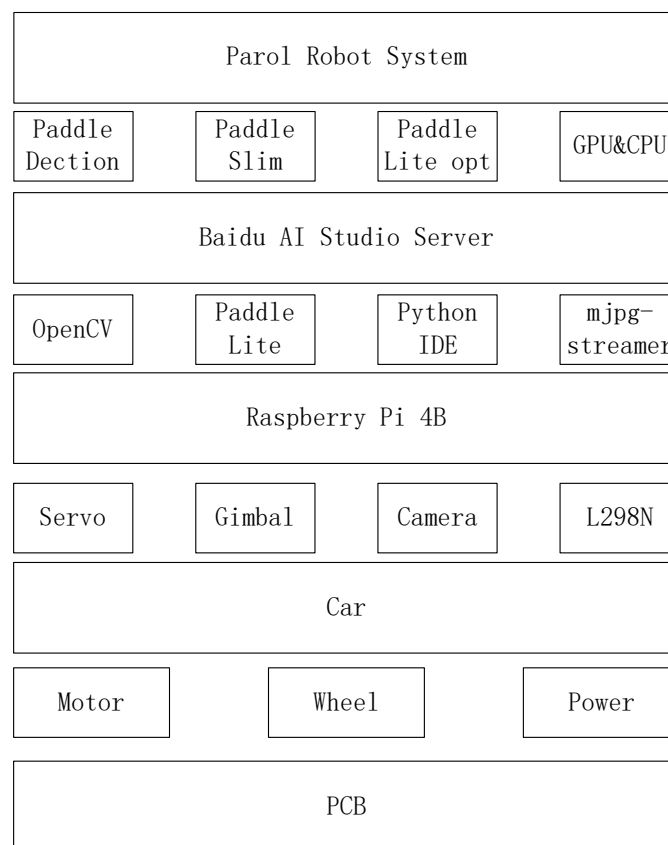


图 1.1 巡检机器人系统架构图

1.3.1 高压线巡检机器人系统设计

本课题研究的机器人系统面向室外环境，主要功能是实现地面平稳行走，从地面对高压电缆进行检测，并将拍摄到的画面实时传至上位机。技术难点是实现机器人地面越障与、高压线异常检测与自动巡线行驶，机器人以四轮驱动小车为原型进行设计，添加相应的外部设备，同时对机器人的环境感知需求进行分析，设计传感系统并进行设备选型，从而进一步完成机器人控制系统的设计。

本课题以四轮直流驱动电机小车为搭载平台，采用光栅传感器和栅格获取直流电机一定时间内转过的角度，从而计算出实时速度和平均速度，进而感知此平台的行驶状态；在此平台上，使用 USB 2.0 摄像头，此摄像头分辨率为 480P，输出格式为 YUYV，置于搭载平台正前方，俯视前方道路，采用 Python3 编程，调用 OpenCV3.2.0 库，将获取的原始图像进行灰度化处理，从而将摄像头的视野变为“黑白”，即将识别出来的黑线、白线用二进制 0 和 1 进行表示，从而将前方道路数值化，再通过设置摄像头识别阈值，进而精确按照设定的颜色（黑色或者白色）行驶；此 USB 摄像头下方是超声波模块，超声波模块用来感知搭载平台和前方障碍物的实时距离，通过所设定的距离阈值来判断前方是否出现障碍物，从而通过舵机转动超声波模块，确认无障碍方向，改变方向，躲避障碍物，继续行驶；搭载平台上方放置主控模块 RaspberryPI 4B，用以实现此平台的全部功能，前方为二自由度舵机云台，此云台用于搭载树莓派 4B 500W 像素摄像头模块，使用 Paddle Lite opt 转化的高压线检测模型和预测程序对获取的原始视频流进行预测，从而识别高压线异物、电工、鸟巢，并通过 mjpg_streamer 插件，将预测视频流传输至网页中，在同一局域网下就在线查看高压线识别状况，二者为该搭载平台的核心模块；另外，本搭载平台的电源模块分为两部分，第一部分是树莓派供电，由于树莓派供电问题一直以来是一个比较棘手的问题，无论是自身供电还是给其他设备供电，故本平台将树莓派和其他模块分开独立供电，第一部分采用充电线和 USB-TypeC 树莓派进行独立供电(5 V 3A)。第二部分的电源模块便是两节 3.7V 可充电锂电池，用于给直流电机、舵机和超声波模块供电，而巡线摄像头和识别摄像头考虑到视频流传输的电磁干扰和接线问题，均有树莓派供电，且实验证明，该两款摄像头不会造成树莓派自身电压不稳定的问题；考虑到本系统的交互功能，特意设置了 RGB 灯，根据系统运行的不同状态和模式，设置不同的发光颜色，便于用户得知当前系统的运行情况；另外考虑到美观和实用性，给每个轮胎加了防护并添加的校徽，更加具有特色。

1.3.2 百度飞桨平台

飞桨公司提供的服务平台包含 EasyDL、AI Studio 和 EasyEdge，它们能够很好地满足各种深度学习对于开发人员的需求^[9]。EasyDL 是专门为大型企业高级用户和系统开发人员提供量身定做的企业 AI 模型设计培训与开发管理咨询服务平台；AI Studio 是针对学习者的在线一体化开发实训平台，集合了 AI 教程、深度学习样例工程、各领域的经典数据集、云端的运算及存储资源、以及比赛平台和社区。EasyEdge 是基于 Paddle Lite 研发的端计算模型生成平台，能够帮助深度学习开发者将自建模型快速部署到设备端^[9]。

本课题主要以 PaddleDetection 为出发方向，在树莓派上部署飞桨的模型压缩工具 PaddleSlim 和轻量化推理引擎 Paddle Lite。PaddleSlim 对模型进行了裁剪、蒸馏和量化，从而直接达到了降低模型尺寸，减少计算机运输耗时的主要目的，Paddle Lite 的高易用性、广泛的硬件支持和领先的性能，可以轻松的实现移动端模型的部署，从而完成整个预警系统的上线部署，最终二者结合可以很好地解决模型难以在小内存设备中运行的问题并保障良好的应用识别效果。

首先采集高压线数据集，标注为 VOC 格式，再使用 PaddleDetection 对数据集和 YOLO V3 DarkNet 模型配置文件对数据集进行训练和评估，从而产出模型文件，在通过预测程序对得出的模型进行测试和优化，待本地 PC 调试完毕后，再使用 Paddle Slim 对模型进行裁剪、蒸馏和量化，从而在不降低检测准确率的情况下，缩小模型体积，便于部署到树莓派当中，然后通过相应 Paddle Lite 对应的 opt 转化模型将本地训练好的模型转化为.nb 文件，并将文件部署到树莓派的 Paddle Lite 当中，再在树莓派编写预测程序，嵌入到 mjpg_streamer 当中，从而将实时检测到的视频流传输到网页当中。

1.3.3 基于深度学习的目标检测算法

相较于其他传统的方法，其中深度的神经网络可以拥有更强的机器学习和计算功能，从而使用户可以实现更加全面的视觉图像特征。它可以分为 two-stage 和 one-stage 两种^[7]，two-stage 顾名思义需要对图像进行两步运算，其中代表性的是 FasterRCNN。two-stage 方法具有高精度，但是难以实现实时检测的效果。然而 one-stage 方法不需要特征区域提取，仅仅一步即可预测最终结果，是一种端到端的目标检测方法，主要代表有 SSD 和 YOLO 系列。其中 YOLOv3 对于检测和检测速度和精度具有良好的效果，将其应用到不同领域成为了研究热点。

目前基于深度学习的目标检测算法，因为其强大泛化能力和学习能力而被广泛应

用。随着目标检测技术的发展，现有的基于深度学习的目标检测算法有以下几类：基于候选框生成检测的 R-CNN、Fast R-CNN、Faster R-CNN；基于回归框检测的方法 YOLO、SSD、YOLO v3、YOLO v4、YOLO v5 等^[10]。这些算法在目标检测的各个方面都具有很高的精度。但针对目标过多、目标相互遮挡、目标尺度过小等因素导致检测准确度下降。为了解决这一问题，决定对 YOLO 系列算法进行研究^[11]，在 YOLO v3 的基础上进行改进^[12]。

最后，针对本研究课题的其他相关技术研究工作项目，还是可能存在一些技术不足，例如关于巡检工作机器人系统续航力和性能的一些问题^[13]、机器人在正常巡检工作的运行过程中主要分别采用了哪种锂电池系统进行动力供电，如何能够确定巡检机器人系统能否在正常巡检的工作过程中能够得以同时具备只有足够的动力电量充足才能直接到达下一级的桅杆塔，或者确定是否搭建能够直接达到下一级的大型太阳能电池充电专用基站，这些都是一个非常迫切需要深入进行研究的课题。除此之外，远程数据库管理系统也需要加入本课题当中^[14]，从而真正的实现高压线远程监控。

2 相关技术概述

本章主要介绍高压线巡检机器人系统的相关技术概念，从机器人搭载平台到树莓派控制，再从树莓派控制到摄像头识别，从摄像头识别到视频传输，最后从视频传输到深度学习和目标检测，即按照感知层、传输层和应用层的顺序对本课题的相关技术进行详细的说明。

2.1 搭载平台

本课题的搭载平台以四轮驱动车为主体，该平台主要用于搭载树莓派、移动电源、直流电机、舵机、HC-SR04 和 USB 摄像头。

2.1.1 电源

电源包含两个部分，一个是充电宝，给树莓派供电，另外一个是一节可充电锂电池，给外部设备供电。

(1) 充电宝

充电宝主要由电芯、电路板和外壳组成。常用的电芯主要有锂和锂聚合物。DC to DC 升压是当前主流升压技术，目前转换效率一般在 60-80%。虽然也有高效率的降压策略，但是对电芯的统一性要求比较高级，故当前基本没有采用该方案的电路设计。

充电宝其实是一款便于随身携带的充电设备，自身将储电、升压和充电集成在了一起，本充电宝输入为 $5V=2.0A \text{ Max}$ ，输出为 $5V=2.1A \text{ Max}$ ，锂电池容量为 60Wh，其工作原理如图所示。

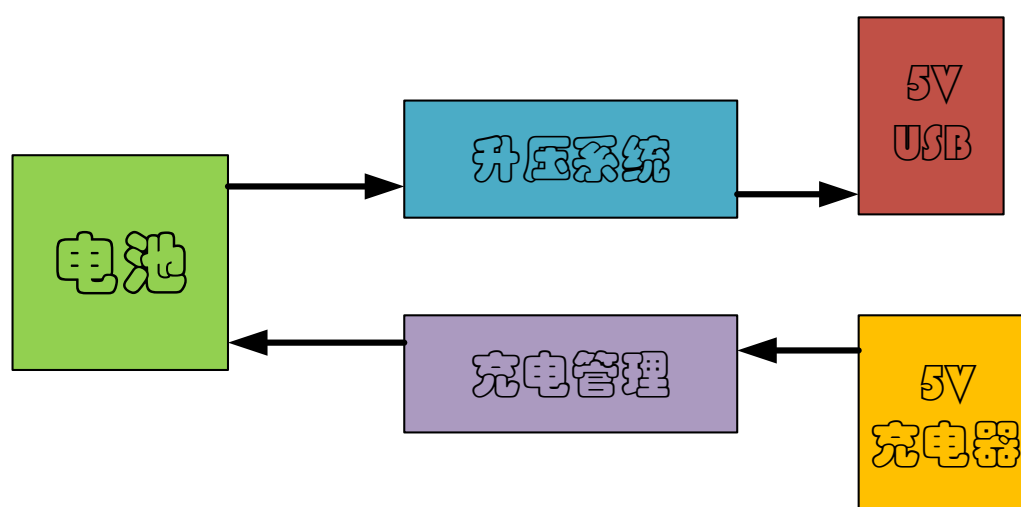


图 2.1 移动电源工作原理

(2) 锂电池

锂电池实际上是可重复使用(可充电)。在充电和放电阶段内,位于层次之间的锂离子脱离所在层次,然后内嵌到两个电极中间。在充电的过程中,位于正电极的锂离子被借用,经过一定时间将电解质溶液插入至正负极中,此时处于正负极的锂离子多于处于正极;放电时间与上述的充电时间完全相反。当今世界上性能最好的锂电池一般均以其锂电极作为核心,工作原理如下图所示。本研究课题所选用的锂电池为 UltraFire HY 18650 7800mAh 3.7V Li-ion。

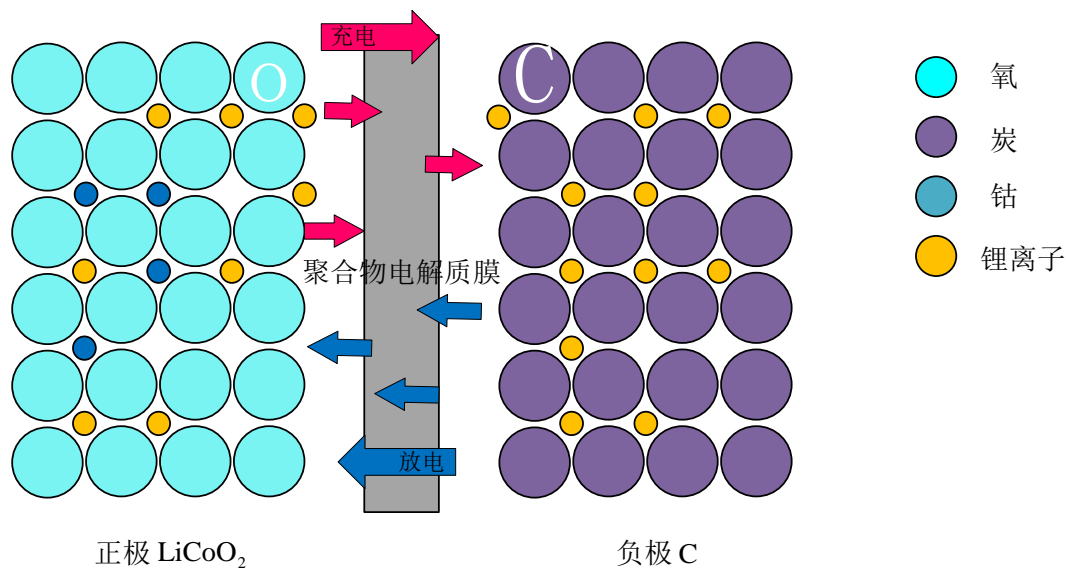


图 2.2 锂电池工作原理

2.1.2 直流电机

直流电机是电能变成机械能的外设,非常适合作为机器人的行走机构。直流电机可承受的电压区间为 1.5V~30V,通过两根杜邦线将引脚连接到电源上就可以使电机开始工作,电压越高,转速越快。交换两根杜邦线的连接顺序就可以直接使电机的反转。该电机转速高、扭矩低,所以作为驱动设备的话,需要外接相应的变速器,变速器可以解决这一问题,进而成功驱动搭载平台。

但是仅有直流电机是不够的,仅靠树莓派的 PWM 信号是无法驱动电机的,还需要电机驱动模块 L298N。L298N 本质是双 H 桥电路,它本身的引脚是电源、控制和输出,其模块引脚说明与工作电路如图所示。

L298N 控制策略和电机的真值表如图所示。如果对电机进行 PWM 调速,需要设置 IN1、IN2 输入的电平信号,从而改变电机的实际转向,之后对使能端输出 PWM 脉冲,从而动态调节电机转速。

表 2.1 L298N 真值状态表

ENA	IN1	IN2	直流电机状态
0	X	X	停止
1/PWM	0	0	停止
1/PWM	0	1	正转
1/PWM	1	0	反转
1/PWM	1	1	停止

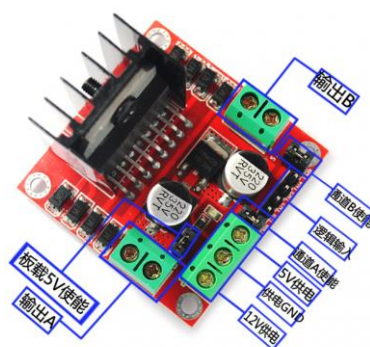


图 2.3 L298N

2.1.3 舵机

伺服电机通常被称为舵机，它是一种带有输出轴的小装置，其内部和外部结构如图所示。当我们给伺服器发送控制信号(高低电平)时，主轴便可以转动到指定角度。输出控制信号只要一直保持原态，伺服电机机构接收到电平信号后，输出轴的角度就不会发生变动。同理，若输出的高低信号电平发生改变，则主轴的角度也会跟着发生相应的变化。由于舵机内置伺服机构和控制电路，尺寸虽小，力度很大，所以舵机在机器人领域中，舵机应用非常广泛。除此之外，舵机能耗和机械负载是呈正相关的。因此，普通的舵机搭载系统并不会需要很多电量。

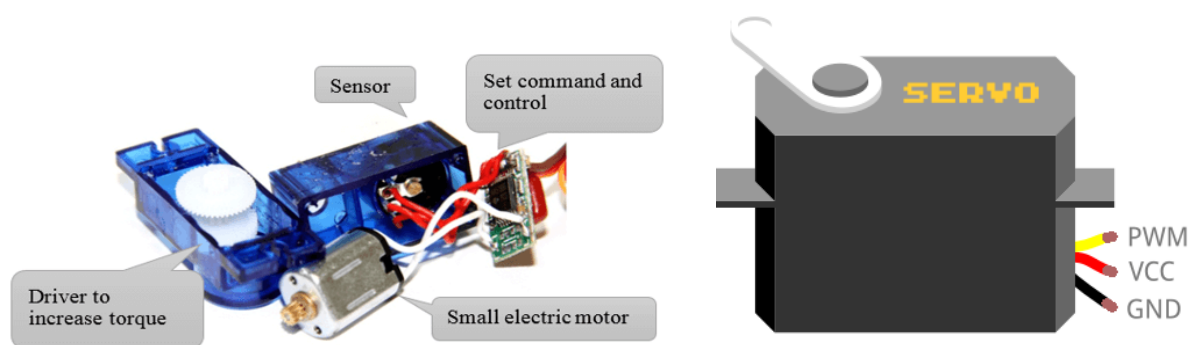


图 2.4 舵机结构

舵机的内部电路、电位设置及驱动器计和发电机都是直接与电路板相互联系的,如上面结构图所示。主控电路通过一个电位计实时监视舵机旋转的角度,若是主轴的位置与高低电平信号相切,则电机就自动停止或者关闭。如果一旦控制电路检测器发现该点的旋转角度不正确,它就可能会停止控制马达的转动,直到其能够达到所规定的旋转角度。舵机功率与和转动角度呈正相关,若轴需转动大角度,电动马达便全力运行,若它只需要转动较小角度,马达便慢速运转,上述过程为舵机转动速度比例控制,即舵机的追随特性。举例如下,设定目前舵机状态位于点 A,此刻主控 CPU 发出一个 PWM 信号,舵机便以全部的速度由从 A 点转至 B 点,此阶段需要花费一定的事件,这样舵机才可以按照指定角度转动到 B 点。

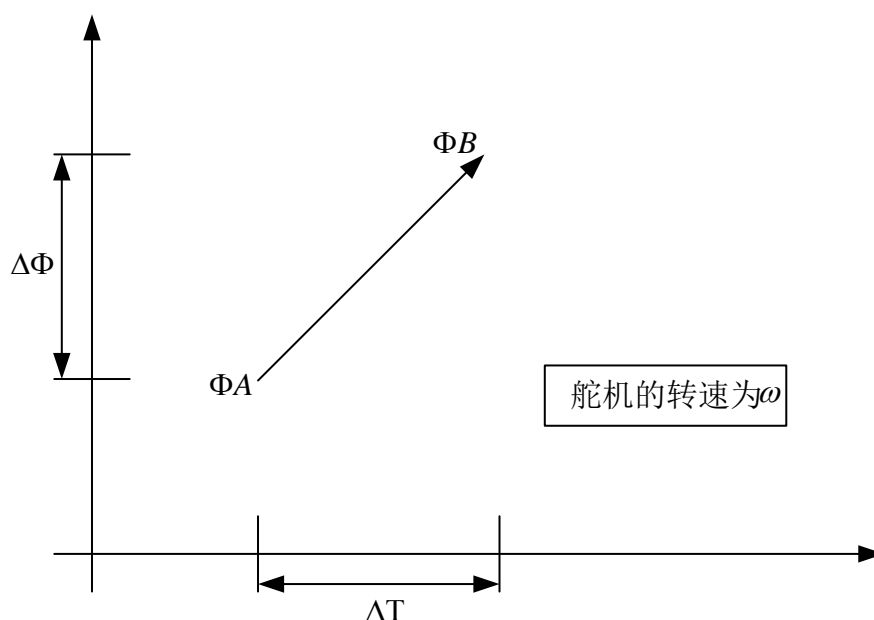


图 2.5 跟随特性

2.1.4 超声波模块

HC-SR04 一般是由一对压电陶瓷超声传感器及其核心的信号处理电路共同构成。这对两个传感器,分别是用来检测发射超声波信号与接收超声波信号。由于放大器发出的信号和它们所接受到的信号强度都比较弱,因此我们需要借助外围的信号放大电路增强它们所发出的信号实际频率,并且在这个时候放大器会把它们转换为高低电平的信号,从而更加稳定地传输至 CPU 进行处理。

首先主控通过 I/O 口输出二进制信号来控制 TRIG 引脚来开始测量距离,此信号为不低于 $10\mu s$ 的高电平信号,然后 HC-SR04 主动发射 8 个 40kHz 的方波,主动判断某时

某刻是否接收到返回的信号。如果接收到返回信号，主控便借助 I/O 口控制 ECHO 引脚输出 1 信号，高电平所维持时长便为超声波发射信号和返回信号的时差，然后得出测试距离 $=(\text{高电平时间} \times \text{声速}(340\text{m/s}))/2$ ，它工作时序如图所示。检测到到 $>10\mu\text{s}$ 的 TTL 信号后就开始使用定时器进行计时，到该引脚变为低电平 0 时就可以获取定时器的原始数值，该数值就为此次测距所用的总时长，然后通过公式就能够得出搭载平台与前方障碍物的实时距离。最后再将上述操作放在死循环中，便可以实施更新每时每刻搭载平台和前方物品的实际距离，一般范围为 1m，单位为 cm。

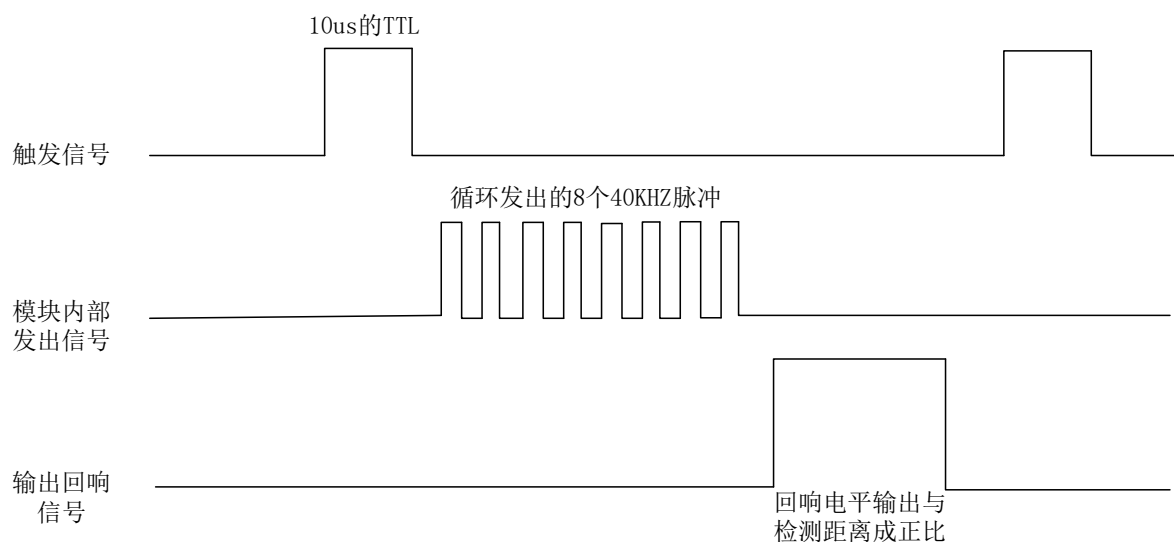


图 2.6 HC-SR04 时序图

2.1.5 OpenCV 巡线

OpenCV 为目前计算机视觉研究领域中经典、实用、强大的开源库，它支持多语言、跨平台。OpenCV 全称是 Open Source Computer Vision Library，其整体框架如下图所示。它拥有整套系列的 API 接口，包括 C、C++、Python、MATLAB 和 Java，可以运行在 Windows、Linux、Mac OS、ios 和 Android 系统上。OpenCV 实现了多种高效且实时的算法，随着 OpenCV 团队夜以继日的开发优化，C/C++接口函数的应用库主要运行在规模较大且 CPU 为多核的大型研发工程当中，从某种程度上来说，这样可以极大提高了计算机视觉算法的应用水平。OpenCV 的研发目的是拼尽全力提升其执行效率。它的编写是使用经过不断优化改善的 C/C++代码，这样才可以完全发挥多核处理器 CPU 的天生优势。OpenCV 的主要设计目的就是搭建适合初级人员开发调用的计算机视觉开源框架，以帮助研发人员可以更加轻松地构造、设计、开发和编写更有深度的计算机视觉相关应用。



OpenCV Conceptual Structure

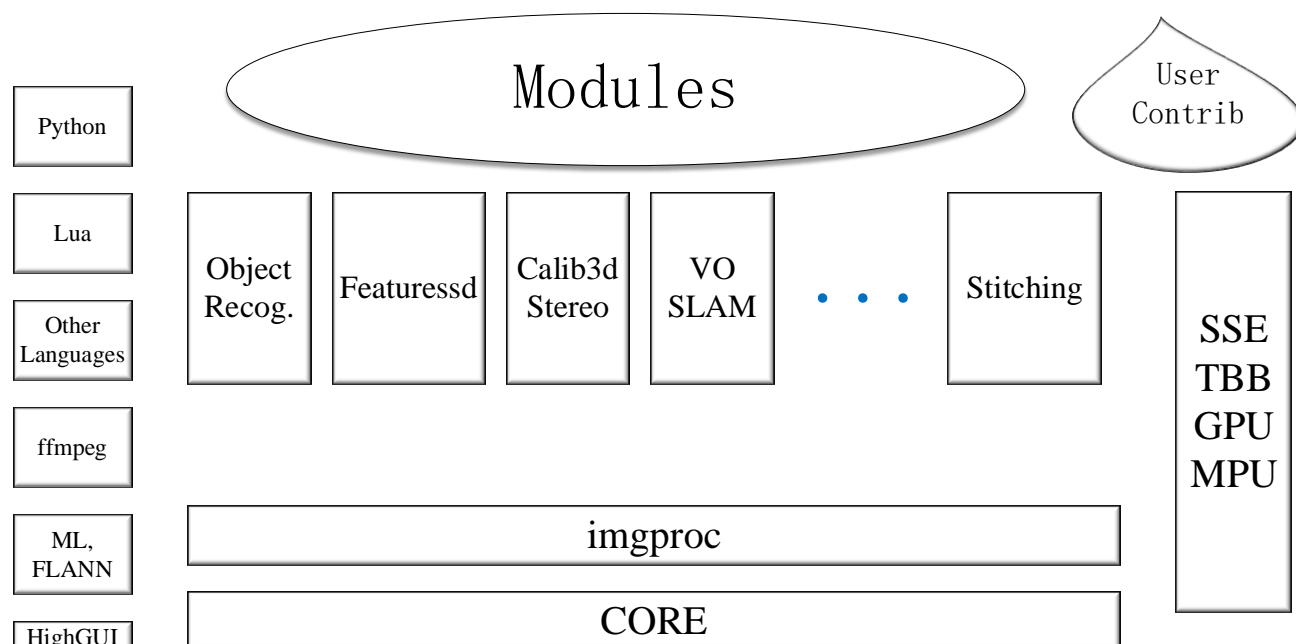


图 2.7 OpenCV 框架

使用该仪器来进行巡线，主要目的就是将 USB 摄像头上所采集到的原始影像数据进行一次二值化操作，OpenCV 提供了多种二值化形式：

`cv2.THRESH_BINARY`

`cv2.THRESH_BINARY_INV`

`cv2.THRESH_TRUNC`

`cv2.THRESH_TOZERO`

`cv2.THRESH_TOZERO_INV`

标准二值化操作就是按照提前设置好的亮度阈值，将超过这个点亮度的节拍给点赋值成白色的 255，低于此点亮度阈值为黑色的 0。目前做巡黑线，使用正常模式 `cv2.THRESH_BINARY`，如果是巡白线，只需要修改成反色模式 `cv2.THRESH_BINARY_INV`。

2.2 Raspberry Pi 4B

树莓派外观如图所示。以小的卡片主板作为存储内存卡的硬盘，卡片的每个主板周围分别各各设有个太网接口和一个以太网通信接口(一般小型机器上没有接入互联网的

接口),可以直接与手机键盘、鼠标和其他互联网通信线路直接相连,同时由自己一人拥有上面的所有卡片核心控制模块均被手工焊接在很小的卡片主控控制板上,具备一切的核心功能。

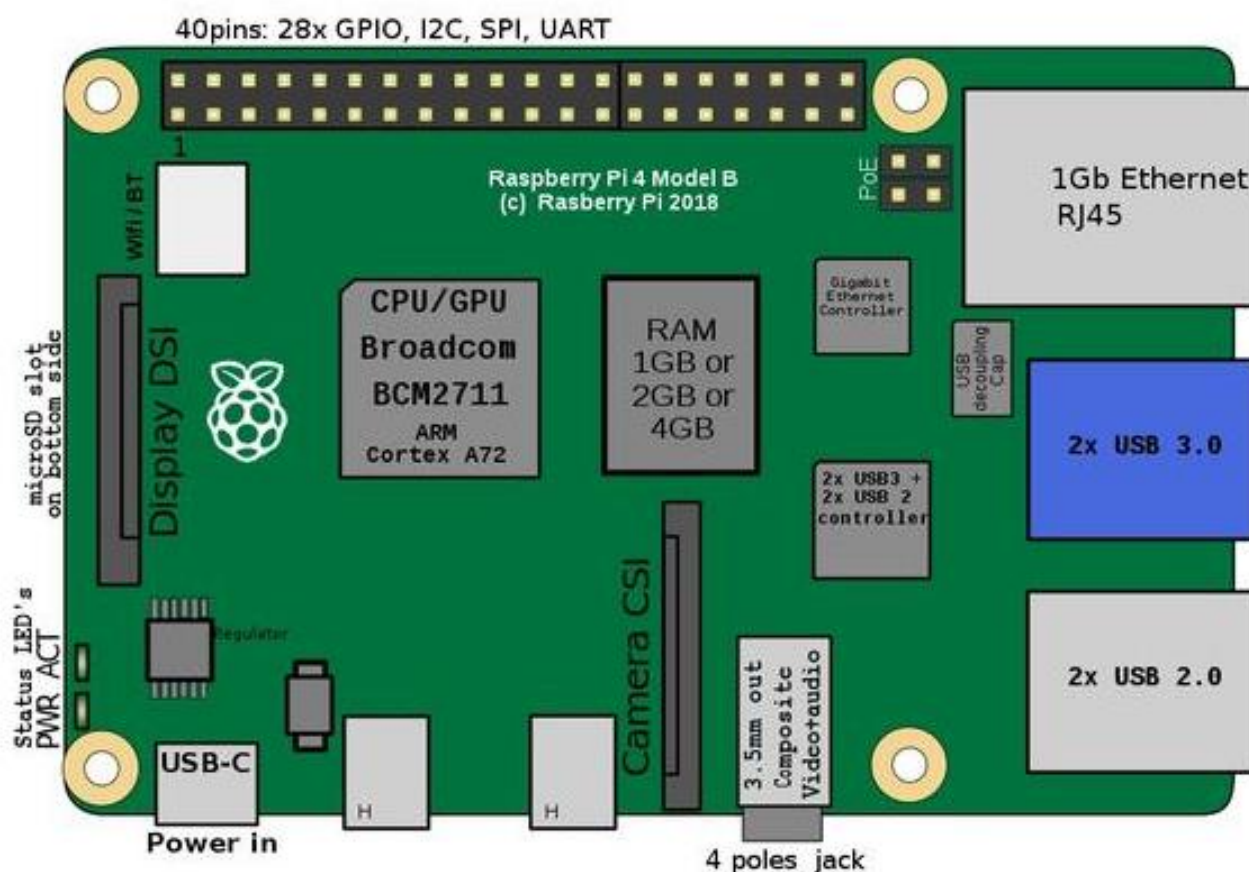


图 2.8 树莓派 4B 接口

树莓派不同于单片机与 Arduino, 其本身自带 Debian-Pi-ARM64 为操作系统, 相当于一个 Mini PC。该树莓派一共有 4 个 CPU 内核, 可同时运行四个进程, 每个进程又可以运行多个线程, 即本身系统具有进程和线程之间的调度能力, 而单片机和 Arduino 需要在工程文件中部署实时操作系统(例如 FreeRTOS、UCOSIII 等小型操作系统)才能够实现多线程任务, 实现难度较大。相比而言, 树莓派更适合多线程编程开发, 主要开发语言为 Python 和 C++, 它不仅可以运行本地编译好的 C++或者 Python 工程, 还可以通过系统命令运行其他脚本文件, 同时执行多个任务, 提升 CPU 的利用率和工作效率。

2.3 Raspberry Pi 4 camera

树莓派 4 代摄像头主要用来对高压线进行异常检测, 此处仅介绍 CSI 摄像头的视频采集技术和工作原理, 后面会介绍相关的目标检测技术。

本课题所采用的树莓派摄像头为 500 万像素索尼 MX219 高清摄像头(可见光),特点为定焦镜头,拍摄可达 3280×2464 像素静态图片、支持 1080p30、720p60、640x480p60/90 摄像功能、扩展板通过上面的 GPIO 插槽连接 Raspberry Pi 主控板,并使用专门的 CSI-2 接口,这是专门为树莓派官方摄像头所研制的适用接口,其与树莓派的连接如图所示。该摄像头默认分辨率为 480P,帧率为 90,通过晶体成像原理将所采集的视频数据通过 CIS-2 接口协议传输至树莓派,树莓派再通过 mjpg_streamer 插件对视频进行处理和推流,进而在树莓派内部实现相应功能。

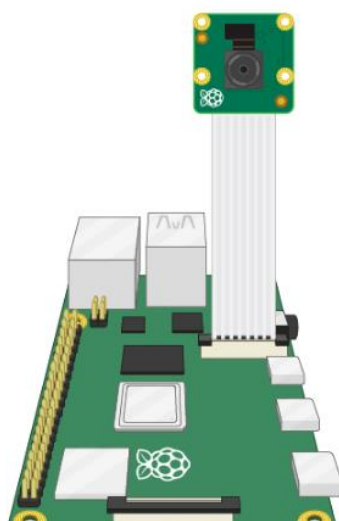


图 2.9 摄像头连接方式

CSI-2 接口规范是由 MIPI 联盟组织于 2005 年发布的相机串行接口, CSI-2 是全新的相机和 CPU 之间的应用接口框架,给便携式、手机摄像头等相关产业提供了一种灵活且高速的设备接口。基于 CSI-2 摄像头数据传输过程使用了数据差分信号对视频中像素值进行传输,同时 CSI-2 传输接口能够非常灵活的进行精简或者扩展,对于接口较少的应用场景, CSI-2 接口可以只使用一组差分数据信号线以及一组差分时钟线就能够完成摄像头的数据串行传输过程,这样便减少了负载,同时也能够满足一定的传输速率,而对于大阵列的 CCD 相机, CSI-2 接口也能够扩展其差分数据线,从而满足多组数据线并行传输的高速要求。同时 CSI-2 接口中也集成了控制接口 CCI, CCI 是一个能够支持 400Khz 传输速率的全双工主从设备通信控制接口,它能够兼容现有很多处理器的 IIC 标准接口,因此可以非常方便地实现 Soc 上 CCI Master Module 到 CSI-2 TX 端 CCI Slave Module 的控制, CSI-2 物理接口框图如下图所示。

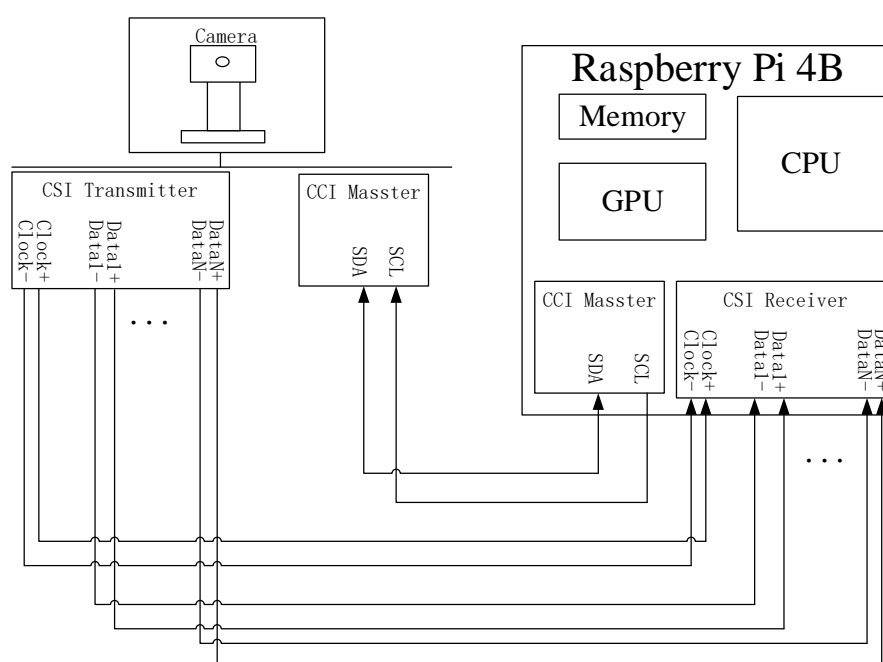


图 2.10 CSI 接口

mjpg-streamer 是开源的并且基于局域网 IP 地址的视频流远程服务器，它可以通过 HTTP 的方式访问 Linux 上面的 CSI 摄像头，从而实现远程监控。mjpg-streamer 是命令行应用程序，它将 JPEG 帧从一个或者多个输入文件复制到多个输出文件。它可以用于通过基于 IP 的网络从网络摄像头流 JPEG 文件到各种类型的视频需求者。本课题的 mjpg-streamer 的调试浏览器便为 Google 浏览器。刚开始 mjpg-streamer 是为 RAM 和 CPU 资源非常受限的嵌入式控制模块研发的。mjpg-streamer 如今支持各种各样的不相同的输入设备。

2.4 深度学习

深度机器学习的这个概念其实就是来源于对人类生物神经系统深度网络的深入钻研，蕴含多个神经隐藏层的多层神经感知器。其中神经网络结构示意图如下。深度学习主要涉及基于卷积运算的神经网络系统、多层神经元的自编码神经网络和以多层自编码神经网络的方式进行预训练以及进一步优化的神经网络权值的深度置信网络(DBN)这三大类方法。因此深度学习常用的模型为 AutoEncoder 自动编码器、Sparse Coding 稀疏编码、Restricted Boltzmann Machine(RBM)限制波尔兹曼机、Deep Belief Networks 深信度网络和 Convolutional Neural Networks 卷积神经网络。

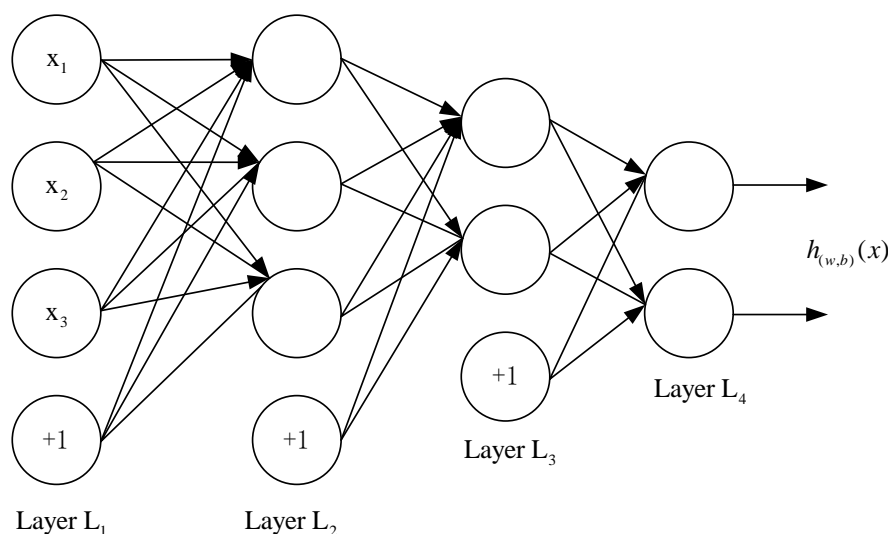


图 2.11 人工神经网络

深度学习的本质是框架主动实现数学建模过程中通用的部分，而开发者只要实现模型个性化的部分即可。在建造数学模型的过程中，每一步需要达成的目标都可分为个性化和通用化两部分。分别由建模者和深度学习框架完成，如表所示。

表 2.2 深度学习过程

思考过程	工作内容	个性化部分	通用部分
	假设一种网络结构	设定网络结构	网络模块和原子函数的实现
Step1 模型设计	设计评价函数(Loss)	指定 Loss 函数	Loss 函数实现
	寻找优化方法	指定优化算法	优化算法实现
Step2 准备数据	准备训练数据	提供数据格式与位置，	为模型批量送入数据
		模型接入数据方式	
Step3 训练设置	训练配置	单机和多机配置	单机到多机的转换、 训练程序的实现
Step4 应用部署	部署应用或测试环境	确定保存模型和	保存模型的实现
		加载模型的环节点	
Step5 模型评估	评估模型效果	指定评估指标	指标实现，图形化工具

目前国内主流的的开源深度学习框架主要包括 TensorFlow、PyTorch 和 PaddlePaddle。虽然 PaddlePaddle 的发展历史没有前面两个深度学习框架的历史悠久，但是它是站在巨人的肩膀上进行开发的，并且百度公司于此之前一直在做人工智能和机器学习相关领域的商业开发，故此研发能力雄厚，PaddlePaddle 迭代、优化次数提升，不断趋于稳定和高效，如今并不逊于 TensorFlow 和 PyTorch。况且由于百度飞桨公司已经通过深度学习框

架与模型实现了高压线异常检测功能，也就是说本课题的研究方案在理论和实践上都是可行的。故本课题的深度学习框架采用 PaddlePaddle，本小节也主要介绍深度学习框架 PaddlePaddle 的相关技术概念。

2.4.1 PaddlePaddle

PaddlePaddle 是指由百度公司官方自主授权开发的一款官方开源、开放的百度讯飞教育桨布式深度教育机器人智能学习应用平台，拥有全面采用官方开源支持的工业级教育应用学习模型，覆盖了 NLP、推荐引擎、计算机视觉等多个领域，并开放多个领先的中文预训练模型，其框架如下图所示。同时它还支持稠密扩散参数和稀疏稠密参数等多场景的大型高效深度机器学习和并行计算技能训练，支持千亿量级规模的稠密参数、数百个或十几点的高效并行计算技能训练，也是一款完全可以直接使用给予学生深度机器学习并行计算相关技术理论基础的培训课程。拥有多端整合部署的开发能力，支持对基于移动端、服务器和云终端等多种不同的异构软硬件预测设备系统进行高速整合演绎，并且在各种目标软件预测系统中的硬件性能上也都具有明显性的优势。

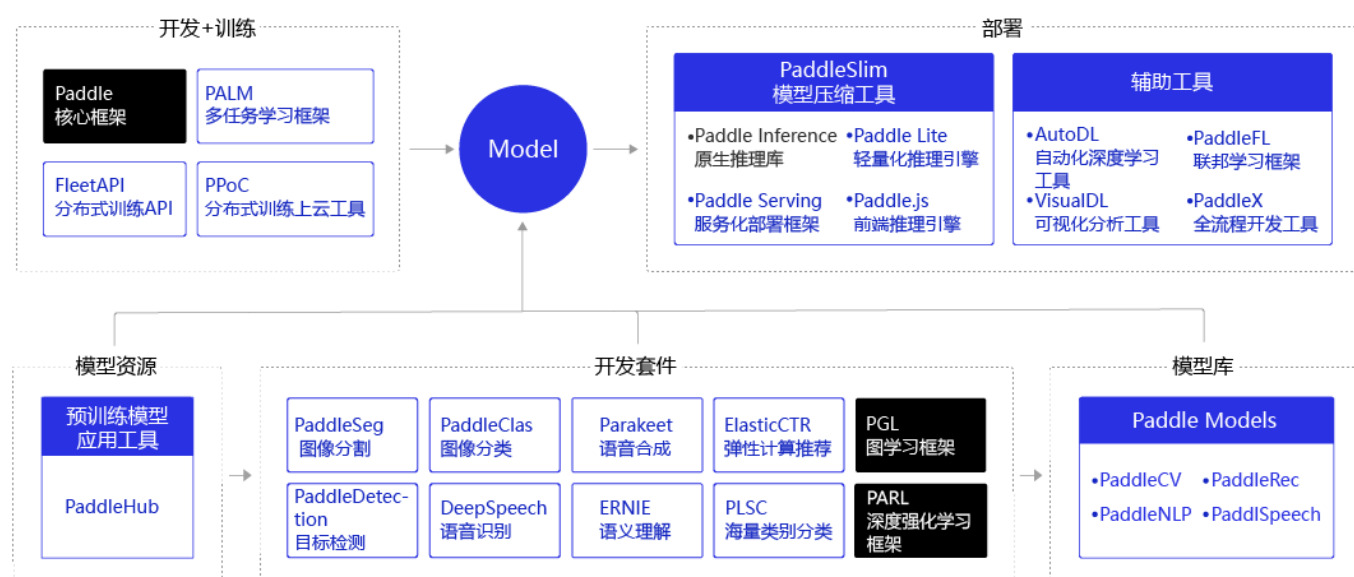


图 2.12 PaddlePaddle 应用架构

PaddlePaddle 核心架构采用分层设计，前端考虑灵活性，以 Python 为主要实现语言，包含了组网 API、IO API、OptimizerAPI 和执行 API 等完备的开发接口；底层框架充分考虑运行性能，因此采用更加偏底层的 C++ 来实现它的功能；内核框架部分主要包含中间表达优化器、存储管理和执行器；内部表示方面包含数据表示（Variable）、计算表示（Operator）和网络表示（ProgramDesc）几个层面。框架向下兼容各种芯片架构，其基

本原理如图所示。

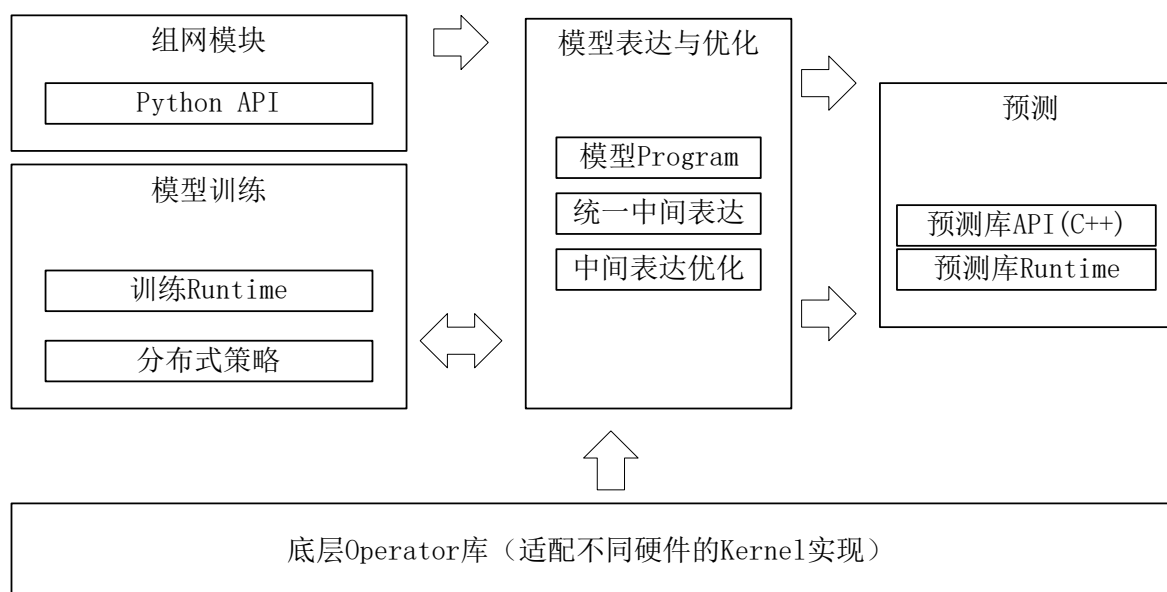


图 2.13 PaddlePaddle 工作原理

2.4.2 PaddleDetection

PaddleDetection 专注于计算机视觉目标检测，是 PaddlePaddle 的一个开发套件，集成了当前市场上主流的计算机视觉识别检测训练模型，方便用户随时进行开发设计，并且可以涵盖当前市场上主流的视觉识别检测算法，即具有高精度的模型、也可以具备高速推理的模型，并为用户提供丰富的预测试训练模型，具有良好的高性能、模块化、产品工业化优点。模块式地进行检测提供，等多种目标检查算法 PaddleDetection 模块化实现多个目标检查算法，设计了大量的扩展数据策略、网络函数等，并且集成了多种跨平台的高性能部署技术和模型压缩。PaddleDetection 模块式地提供 YOLOv3, EfficientDet 等 10 余种目标检测算法、ResNet-vd 等 10 余种 backbone，以及 sync batch norm, IoU Loss、可变量卷积等多个扩展模块，这些模块可以灵活配置、自由组合，同时预置提供 100 余种训练好的检测模型。

2.4.3 PaddleSlim

因为神经网络计算难度和复杂度较高，模型计算中的一些参数数据储存量大，限制了一些商业场景和应用设备上的软件安装和设备部署，尤其特别适合便携式的 pc 和嵌入式应用设备。所以模型的应用小型化设计技术也从早期最初一代开始的例如参数模型量化、知识模型蒸馏、网络模型修饰和小剪枝等逐步逐渐发展普及到最新一代神经网络的基础架构例如搜索 (NAS) 和自动模型压缩等。与此同时，PaddlePaddle 自动模

型蒸馏压缩工具 PaddleSlim 应运而生，PaddleSlim 除了可以允许系统使用者在原生态中基于传统的神经网络模型剪枝、参数模型测量化及相关专业知识进行蒸馏等压缩方法外，还提供可以同时支持当前最新型的神经网络结构及自动模型进行压缩。

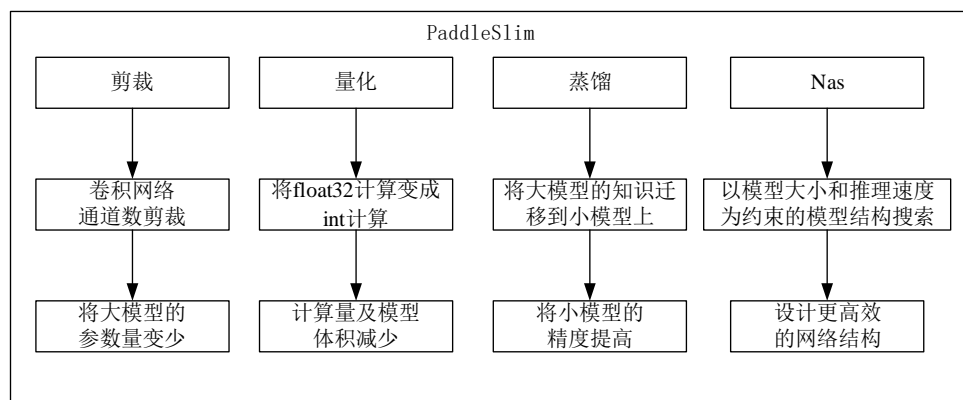


图 2.14 PaddleSlime 框架

2.4.4 Paddle Lite

PaddleLite 支持 ARM CPU、ARM GPU、Huawei NPU、Intel X86 CPU、NV GPU 等移动嵌入式设备的部署安装。得益于对不同硬件适度的抽象，在 Paddle Lite 框架本身清晰的同时支持不同硬件的特殊调度需求，使得 Paddle Lite 架构在框架的清晰程度和硬件的特定调度优化上达到很好的平衡，而且多种硬件的 Kernel 在代码层和执行层均互不干扰，尽管图优化上有复杂的策略，但并不影响移动端的轻量级部署，图分析模块和最终的执行引擎可以拆开使用，最终部署只有一层薄薄的 Kernel。Paddle Lite 流程如图所示。

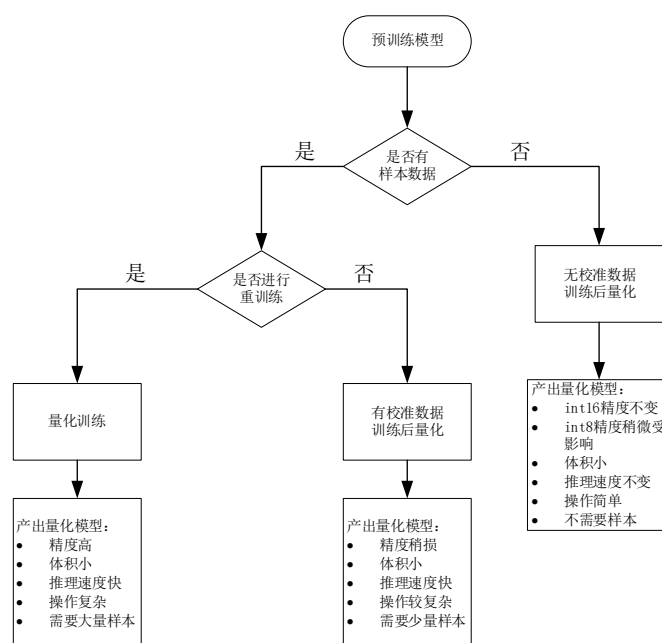


图 2.15 Paddlelite 工作流程

Paddle Lite 架构主要侧重于对多硬件、高性能的支持，其主要的设计理念如下，引入 Type System，强化 Data Layout 的混合调度功能、多硬件、量化的方法；硬件细节进行隔离，通过不同的编译开关，对于支持的任何一个硬件都可以自由地插拔；引入 MIR(Machine IR) 的概念，强化带执行环境下的优化支持；图优化模块和执行引擎实现了良好的解耦拆分，保证预测执行阶段的轻量和高效率，架构图如下。

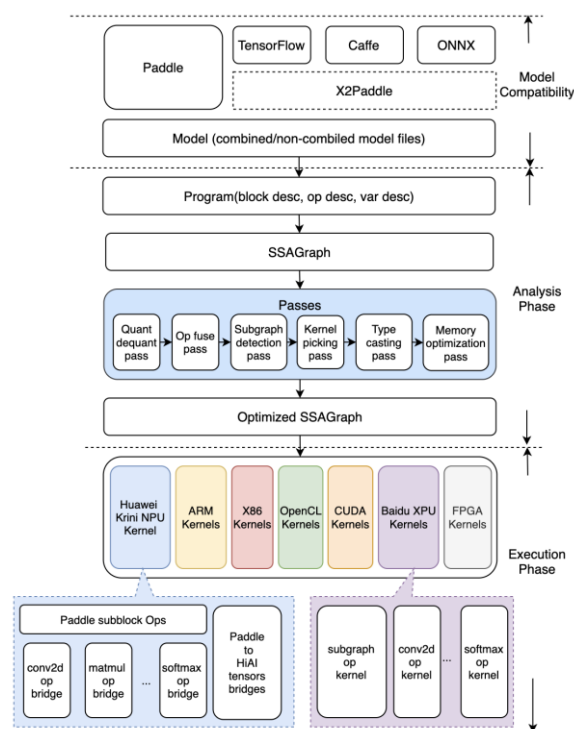


图 2.16 Paddle-Lite 架构

2.5 目标检测-YOLOv3

在深度学习中对目标进行检测的算法中，基于锚框 (Anchor) 检测的方法主要可以划分为两个阶段性的方法和第一个阶段性的方法。两个阶段的方法先对自己感兴趣的地点进行选择，然后再对候选框内的地点做出分类和回归。两阶段的模型有 R-CNN 系列，比如 R-CNN，Fast-RCNN 等。两阶段模型的优点是精度高，但是速度及较慢。一阶段方法直接对 Anchor 进行分类和回归，得到最终类别和目标框一阶段模型的推理速度较快，但是相对的精度会有所下降。YOLOv3 模型设计思想如下：

YOLOv3^[15]模型系统首先通过利用维度集群作为一个锚框模型来预测各维度的边界框，使用一定的逻辑回归方法来预测各维度的客观评价分数。然后通过使用一种逻辑式的分类器将多个标签进行了分类，进而预测了边界框中各种可能被包含的物体类别，并且在训练的过程中通过使用了二元交叉熵的损失方法进行了分类预测，对于种不同尺

度的物体边框进行了预测，该系统利用与金字塔网络相似的理论概念从这些物体尺度中自动提取出其特征。YOLOv3 使用一种新的网络进行特征提取，该网络是 YOLOv2、Darknet-19 中使用的网络和新遗留网络之间的混合方法，即 Darknet-53，然后使用多尺度训练、批量归一化与标准化，最后使用 Daeknet 神经网络框架对数据进行训练和测试。

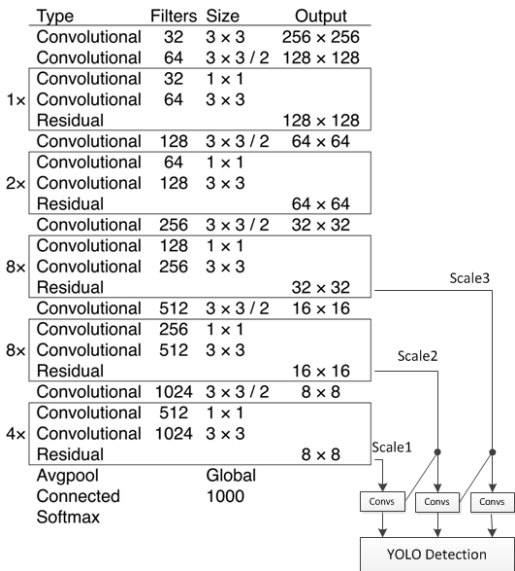


图 2.17 Darknet 神经网络框架^[15]

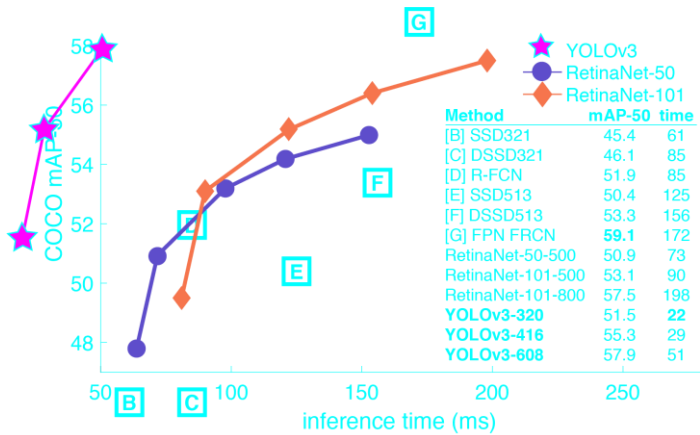


图 2.18 YOLOv3 测试^[15]

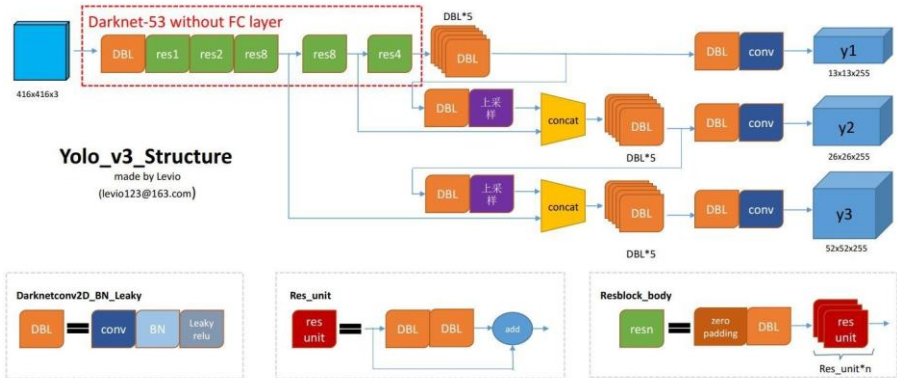


图 2.19 YOLOv3 框架^[15]

2.6 本章小结

本章按照自下而上、由内而外的顺序对本课题所研究设计的内容的各个部分都进行了相关技术的阐述，无论是底层开发，还是数据传输，或者上层应用，都介绍了他们的概念、工作原理和实际用途，并且这些相关概念都在本课题中有所体现。

3 高压线巡检机器人系统设计

本章主要介绍高压线巡检机器人搭载平台的结构设计、外部设备和控制模块的搭建过程以及平台基础功能的实现方案和策略。首先根据当前各种巡检机器人的结构设计和周边可用资源情况对搭载平台结构进行选型和制作，然后根据机器人平台运动、感知等各个方面的需求对外部设备模块进行选型、学习和安装。布置好整体结构与外部设备之后，便开始设计本课题搭载平台的控制部分，首先是电路设计，根据每个模块的工作原理和安装位置，焊接电路转接板，并使用杜邦线将各个模块、转接电路板和树莓派 GPIO 口进行连接，最后根据前面已经设计的系统方案，对每一个功能进行模块化编程，最后对他们进行整合、调试和优化，最终完成该巡检系统结构的设计和基础功能的实现，接下来对上述各个功能部分进行详细叙述和设计。

3.1 机械系统设计

3.1.1 需求分析

目前高空运输电缆线路的巡检技术方式主要可以分为两种航测法和机器人巡检两种方法，其中航测法又主要包括了无人机巡检和直升飞行员巡检，同样，机器人巡检又主要包括地面巡检和高空巡检。第一章现状分析已经阐述为何选择机器人巡检法中的地面巡检，此处不在赘述，仅介绍以地面巡检为原型的机械系统设计。

对于地面巡检，工业中主要以装甲车为检测平台载体，以人工地面目测为原型进行功能开发。装甲车即可由工作人员操控行驶，也可根据路线规划或者电磁线路自动行驶，其上载有三自由度或者二自由度工业相机云台，对高空电缆进行实时监测和识别。根据工业中地面巡检机器人的结构设计，并结合自身资源和身边环境，决定以四轮驱动小车为该巡检机器人的搭载平台，并搭建安装二自由度舵机云台。

3.1.2 详细设计

本搭载平台的机械系统设计材料主要为亚克力板、电路板、热熔胶、铜柱、硬纸片等原体固定材料。用亚克力板设计车身，共分为三层，底层用于安装直流电机、轮胎，分别安装至底层亚克力板的边缘凹槽中，并用铜柱和螺母进行固定。此外，底层用于放置可充电锂电池供电模块、舵机、直流电机独立控制开关模块、超声波搭载舵机模块以及必要电源线和信号连接线；中层为核心电路部分，其构造与底层亚克力一致，二者位

置为平行对应关系，主要用于放置直流电机 L298N 驱动模块、5V 可调节稳压模块、GPIO 转接板和移动电源，这些核心部分是底层设备和上层设备通信的关键中间组件，同时也是整车架构的关键支撑组件；上层为核心控制模块，用于放置 Type-C 供电线、主控模块-树莓派 4B 和载有树莓派四代 CSI 接口摄像头的二自由度云台。三层之间通过定长铜柱、螺丝、螺母进行连接、支撑和固定，而每层之上，使用上述连接装置、热熔胶、胶带、扎带对相关设备进行固定，并用杜邦线连接各个设备之间的信号接口。另外，考虑到四轮驱动系统在地面行走时，其轮胎会使碾压的细小异物飞落在中层上面，由于中层是关键部分，要对其进行保护，防止异物对线路造成短路，出现不可逆的永久损伤，故用硬纸片分别将轮胎进行包装和防护，这样即保护了轮胎，又保护了中层设备。最后，考虑到整车的美观性，对线路和设备安装进行了迭代优化，并将校徽放置在延申至中层的轮胎防护片上，这样一来，即达到了实用性，又达到了美观性，体现了产品设计的重要思想。

3.1.3 关键技术

本部分的关键技术为线路和设备在三层之间的布局优化，从开始制作到开发完成，此搭载平台的结构设计共迭代了 5 代。起初整车结构仅有两层，仅有上层（当前中层）承载的所有模块，主要是移动电源、可充电锂电池和树莓派，三者极大的消耗了整车体及和容量，严重影响了线路布置和美观性，而且一旦出现不慎操作，即不利于维修也不利于安装布置其他模块，于是又添加了上层载体，从而形成两个空心空间用于安放关键模块（对整个系统起着举足轻重的作用，并且需要保护，最底层和最上层分别放置电机和核心控制检测模块，这样不仅扩大了使用空间，还增加了整车的稳定性）。

3.2 控制系统设计

3.2.1 底盘模块

该部分主要有 4 个小型 5V 直流电机和 L298N 直流电机驱动组成，其电路连接如图所示。此模块通过 Motor.py 程序实现，共有前进、后退、左转、右转、原地左转和原地右转六种运动状态。树莓派 GPIO 口将高低电平信号发送至 L298 的 IN1~IN4，分别控制左前方、左后方、右前方和右后方轮子的转动方向，从而改变运动状态，其 GPIO 输出信号和运动状态的对应关系如下表所示。PWM 波频率为 2KHz，周期为 0.5ms，初始占空比为 0，最大占空比为 100，默认占空比为 80，但 PWM 占空比实际上不能大于 80，因为大于 80 之后，树莓派的摄像头由于其行驶过快造成视频传输丢帧。

表 3.1 电机控制信号

GPIO(20 21 19 26)	IN1~4	Status
1010	1010	ahead
0101	0010	rear
0010	0010	left
1000	1000	right
0110	0110	spin_left
1001	1001	spin_right

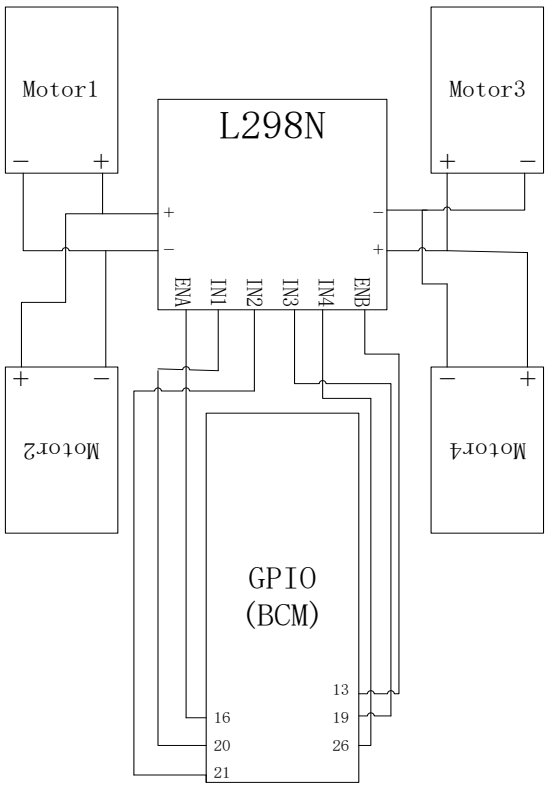


图 3.1 电机连线

3. 2. 2 避障模块

超声波模块采用 HC-SR04，树莓派主要通过 GPIO 控制 Trig 引脚发送超声波，检测 Echo 是否接收到反射回来的超声波，然后根据 HC-SR04 的工作时序周期性的发送控制信号，从而获取实际距离。该模块主要为自动避障所服务，其避障策略如下图所示。常规避障策略是通过三个甚至四个超声波模块检测多个方位障碍物情况，进而做出判断，重新规划行驶方向。而本机器人系统仅通过一个 MG90S 舵机和 HC-SR04 即可实现，当前方检测到障碍物时，便转动舵机寻找没有障碍物的方位并继续行驶，当再次检测到障

碍物时，舵机转动方向与上次相反，但是逻辑处理顺序相同，仍是寻找可以继续行驶的方向，只是起始寻找方向相反，这样设计的目的是防止机器人原地转圈。

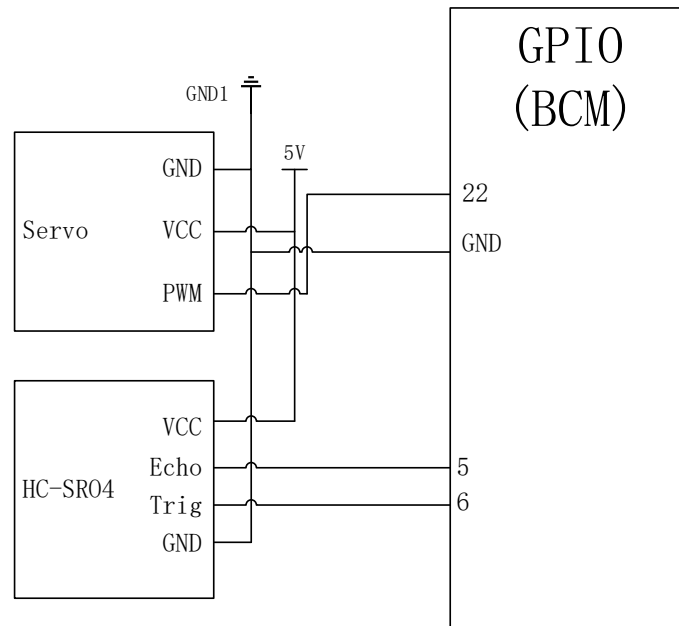


图 3.2 舵机和 HC-SR04 连线

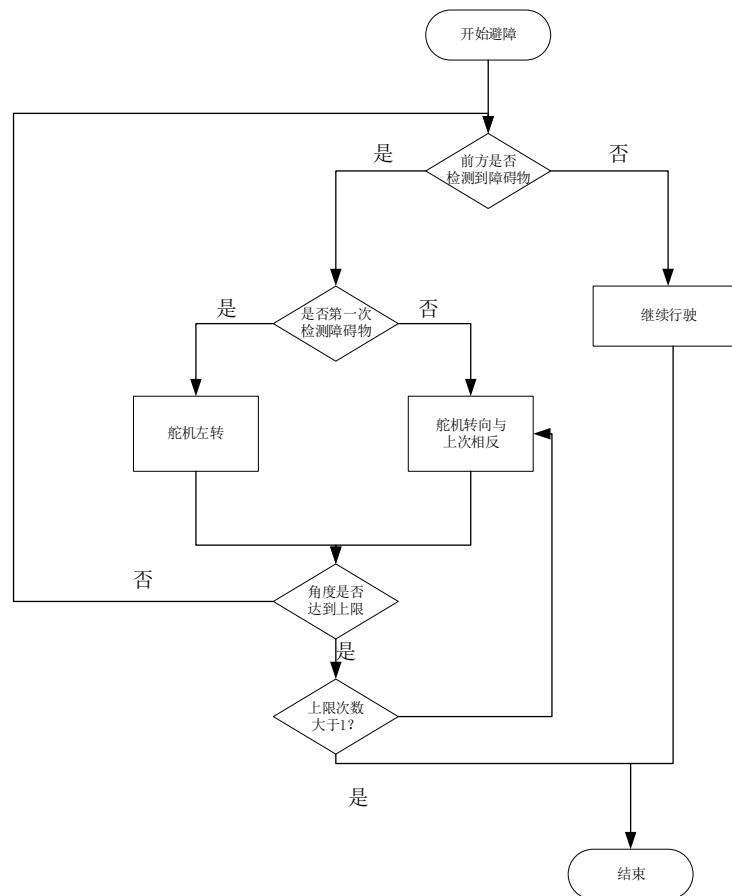


图 3.3 避障流程图

3.2.3 巡线模块

巡线模块采用 USB2.0 480P 摄像头进行巡线，依赖库为 OpenCV，编程语言为 Python3，服务目的主要是实现机器人搭载平台的自动地线巡检功能，相比较长时间人工驾驶巡检车辆，机器人搭载平台自动行驶是非常重要的，因此设计了该模块，此模块和超声波避障模块共同使用来模拟工业中巡检机器人的地线自动巡检。其主要思路如下：

OpenCV 库调用 USB 摄像头获取原始图像，并通过 cvColor 方法对图像进行灰度化处理，然后通过 threshold 方法对灰度化后的图像进行二值化，再根据实际调试情况，对二值化后的数据进行膨胀（dilate）和腐蚀（erode），上述方法也叫边缘检测算法，通过边缘检测算法确定黑线中间点位置，再将其与目标点进行比较，从而获取偏移量，即阈值和实际值之差，树莓派再根据偏移量大小计算出期望值，再将期望值作为信号发送至 L298N 模块，进而调整轮胎转向和转动速度，从而实现自动巡线行驶，算法流程图如下。

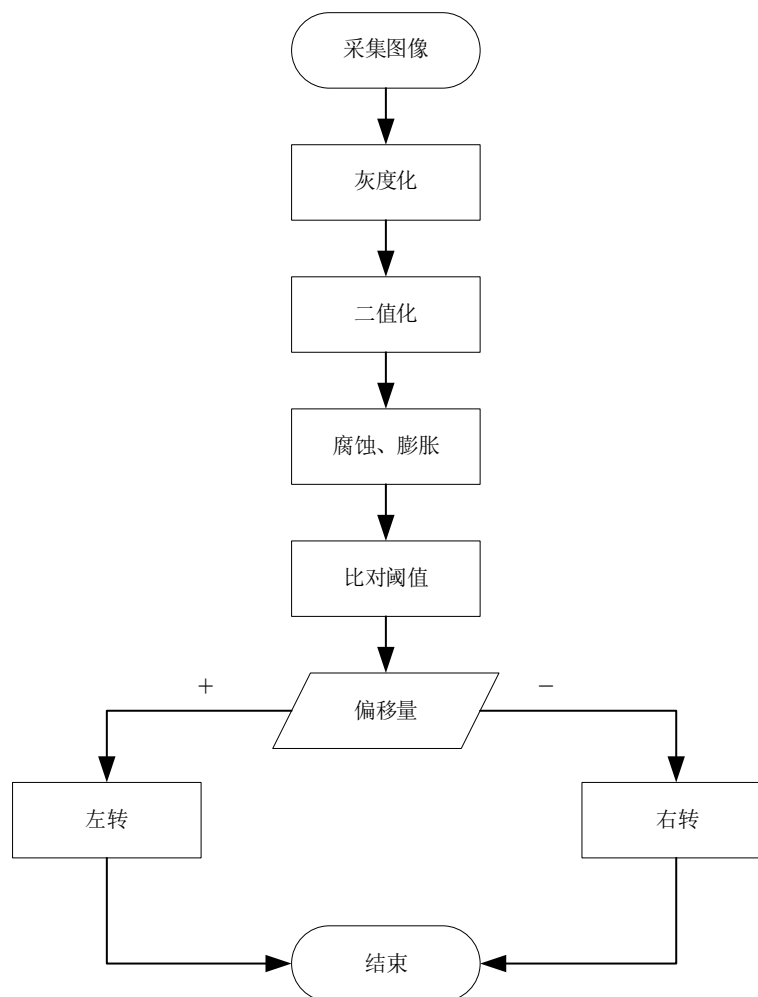


图 3.4 循迹流程图

3.2.4 键盘模块

本模块主要功能是使用 2.4Ghz 无线键盘或者 SSH 连接的本地键盘来远程操控巡检机器人，本模块由 Python3 实现，依赖库为 pygame，pygame 本来是为制作页面游戏服务，但其封装好的键盘事件监听库能够很好满足巡检机器人远程控制需求。本项目通过 pygame 线程来实现键盘按键监听事件，键盘控制按键说明请见附录 1。

首先通过 pygame.init()和 pygame.display.set_mode()操作初始化键盘操作页面，只有当焦点聚集在窗口内部才可进行控制（此处不必考虑，当脚本设置为开机自启动时，焦点只会在操作窗口内部），通过监听键盘按下事件来连续控制底盘电机、云台舵机和超声波舵机的转动，通过监听键盘按下抬起操作来控制机器人运行模式，例如人工驾驶、自动行驶、关闭系统、重启系统等操作，其控制流程图如下。

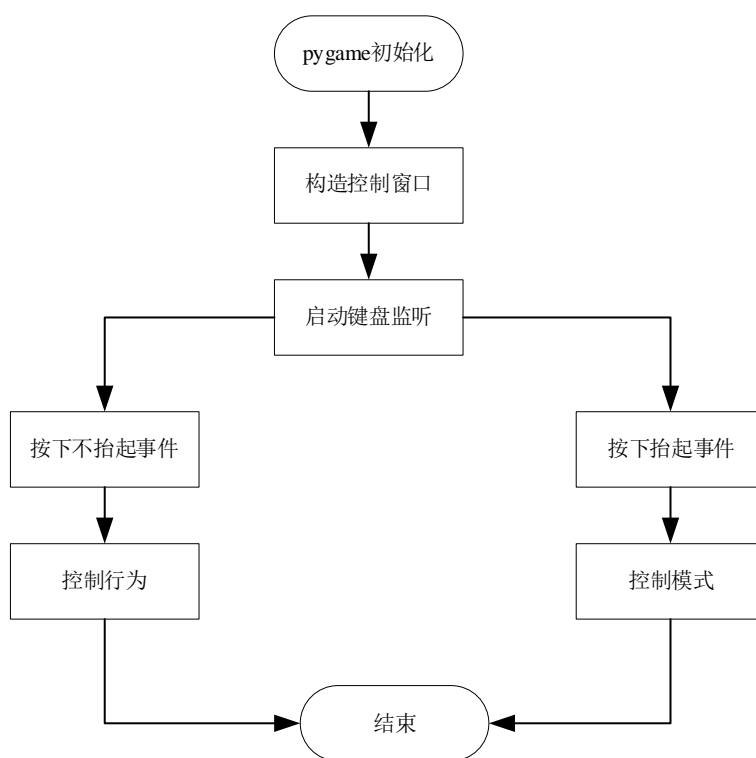


图 3.5 pygame 流程图

3.2.5 舵机模块

本模块共有三个舵机，包括两个 TS90A、一个 MG90S 舵机，前者构成二自由度云台搭载树莓派 4 代摄像头，用来调整摄像头检测视野，提升检测的灵活性，后者用于搭载 HC-SR04、USB 巡线摄像头模块和交互模块 RGB 彩灯，由于前方和上方视野可通过树莓派摄像头获取，故仅需一个 MG90S 舵机实现二维平面的转动，且具有输出小、力矩大，能够满足当前机器人运动要求，三个舵机与树莓派的连线如下图所示。

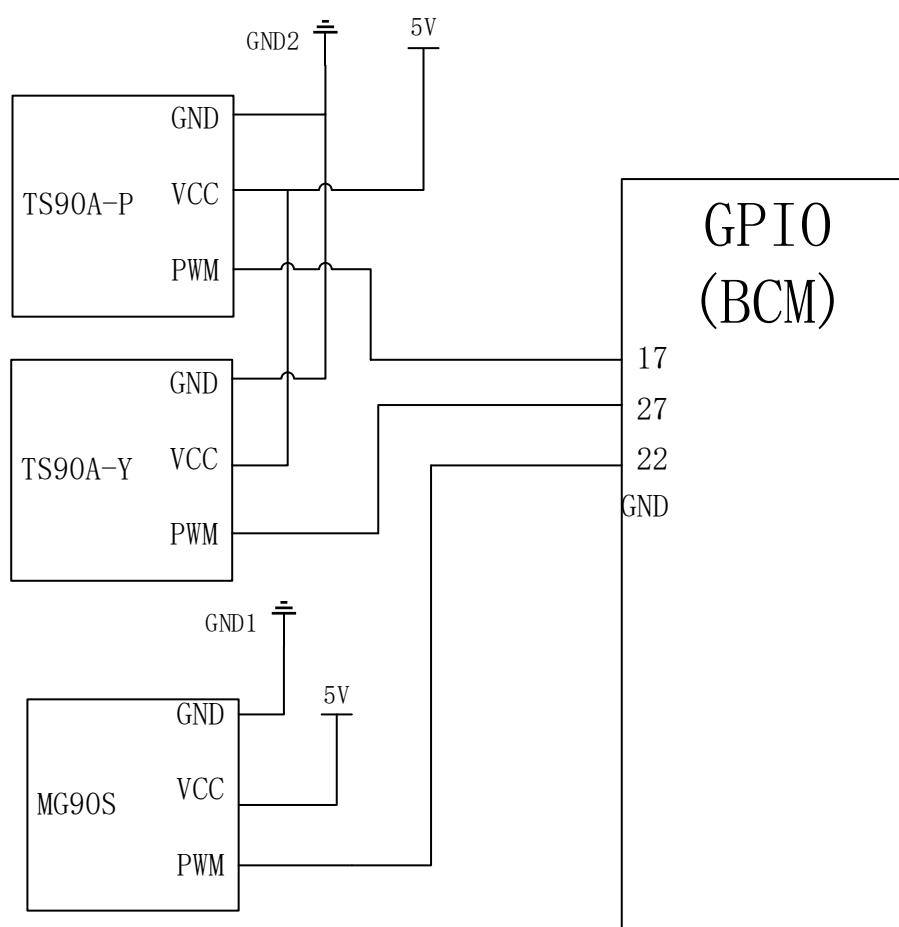


图 3.6 舵机连线

舵机控制程序中共有四个方法，分别为正转（`positiveRotation`）、反转（`reverseRotation`）、定转（`specifyRotation`）和归中（`clear`）。正转和反转是通过在周期时间内连续给定一定的步进角度来实现连续转动，此部分需要结合按键模块中按键按下监听事件来实现，定转便是通过一定的 PWM 波输出规定的占空比，使舵机直接转动到相应角度，但需要一定的延时时间，仅为舵机归中功能服务，归中便是将舵机角度恢复到初始化状态。舵机的控制还需要进行消抖，不然每次输出信号后，舵机都会不断抖动，方法便是再输出 PWM 信号后，再输出归零信号，但二者之间需要时差，否则舵机会在原始位置不断抖动，这样一来，便成功解决了舵机抖动问题。

3.2.6 交互模块

任何设计的产品都要考虑与用户之间的交互，本交互模块完成之前共有两种选型，分别为 OLED 显示屏和 RGB 彩灯，虽然 OLED 可以显示更多信息，但驱动相比 RGB 等来说比较复杂，需要对屏幕不断刷新，占用 CPU，况且实际用途中，OLED 没有过多的用途，相反 RGB 彩灯可以出现多种颜色，只要每种颜色对应机器人的每种状态便可

实现人机交互，且不需要实时刷新颜色，其颜色与状态对应表如下。

表 3.2 RGB 指示灯

颜色	状态	备注
红色	上电初始化	中途变红表示模块异常
青色	外设初始化完成	
蓝色	一切准备就绪	等待其他指令操作
绿色	手动模式	0
黄色	自动模式	1
青色闪烁	正在测距	‘-‘控制
紫色亮起	检测到目标	常亮为巡线模式

3. 2. 7 关键技术

本部分的关键技术为巡线和多线程控制。巡线中最关键的是对二值化后的图像进行腐蚀和膨胀，从而找出白线或者黑线的中间点，并进行对比获取偏移量，根据偏移量调节四轮转动方向和转动速度，具体内容本部分的第 3 节已进行阐述，此处不再赘述。

本部分的程序框架如下图所示，所有的模块都为主程序文件 `raspicar.py` 服务，由于在一个 `while True` 的死循环中实现所有模块的功能仅靠一个线程来实现，无法提升单个 CPU 的利用率，况且每个模块的访问方式为轮询访问，造成某些关键模块不能及时调用，因此需要为每一模块分配一个线程，这些线程可以同时运行在 GIL 锁机制下的单 CPU 中，虽然这些线程只能通过一个 CPU 来运行，无法使用树莓派另外三个 CPU，但是机器人系统属于 GPIO 密集型任务，一个进程多个线程的运行方式完全可以满足需求，提升程序运行效率，另外，树莓派本身就是一个操作系统，接下来的基于深度学习的目标检测算法单独使用 3 个 CPU 来运行，这样一来，便合理分配了树莓派的 4 个 CPU，也充分发挥的树莓派 4B 的运行处理能力。除此之外，还有一个 `while True` 死循环和主程序用来控制线程的开启和关闭以及 RGB 彩灯的检测和指示更新。

一共为四个线程，分别是控制模式的 `Thread_Control`、控制底盘的 `Thread_Motor`、控制探测的 `Thread_Detect` 以及控制二自由度云台的 `Thread_Steer`。`Thread_Control` 线程用于监听键盘事件控制直流电机，也可根据采集的偏移量和距离来改变运动参数；`Thread_Steer` 线程用于控制舵机云台的运动；`Thread_Detect` 线程用于控制 HC-SR04、USB 摄像头和舵机的运行；而 `Thread_Control` 线程主要通过键盘按下抬起事件来控制巡

检系统的运行状态。四个线程运行过程都会上锁，且共享 Shared 类中的所有成员，互不干扰的同时又能相互调度。

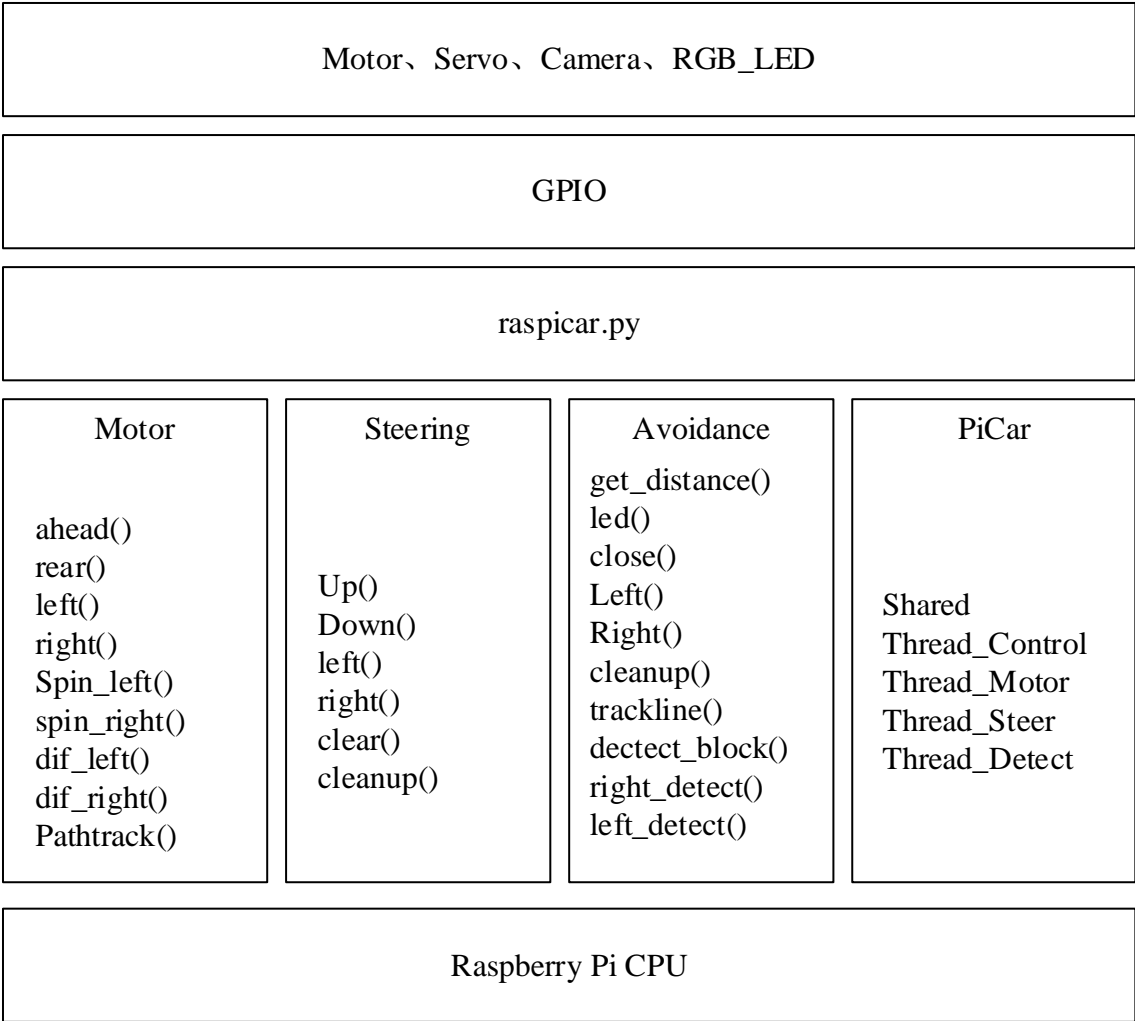


图 3.7 系统程序架构

3.3 本章小结

本章在对机械结构进行需求分析之后，分别阐释了底层、中层、顶层的构造、功能和联系，并介绍了整车的结构优化和美观设计方案。在完成搭载平台的结构设计之后，按照外设模块的顺序依次叙述该系统实现程序的算法思路和运行逻辑，最后阐述了多线程编程和 OpenCV 巡线关键技术，总体来说，自下而上的介绍了整个巡检机器人系统的设计、编程和实现。

4 高压线异常检测研究

巡检机器人在地面行走时还需要对高压线进行异常检测，主要检测对象为人、鸟巢和缠绕物。本章主要介绍服务器端模型的训练、评估和预测以及 opt 模型转化，树莓派端 Paddle Lite 的部署、训练模型的预测以及实时预测视频流的传输。

4.1 实验环境

数据标注在本地笔记本电脑运行，操作系统为 Win10，语言为 Python3.7，标注工具为 Anaconda 的 labelImg，标注格式为 VOC，将标注好的数据集送至 BaiDu AI Studio 服务器进行处理；服务器的操作系统为 Ubuntu18.04，GPU 为 Tesla V100，Video Mem 为 16GB，4 核 CPU，RAM 为 32GB，磁盘大小为 100GB，python 环境为 3.7.3，其余组件版本信息如上表所示；树莓派操作系统为 raspi-buster-arm64，python 环境为 3.7.3，OpenCV 版本为 3.2.0。具体版本信息如下表所示。

表 4.1 版本信息

组件	版本	系统	编程语言
paddlepaddle	2.0.2		
paddleDetection	2.0.0	Ubuntu18.04	Python3.7.3
paddleSlim	2.0.0		
paddleLite	2.8		
OpenCV	3.2.0	Raspi-buster-arm64	C++

4.2 模型训练

本部分在 PC 端完成，主要包括数据采集和标注，然后使用 paddleDetection 并结合标注的 VOC 数据集和 YOLOv3 Darknet53 模型配置文件进行训练、评估和预测，从而导出最终模型。

4.2.1 数据标注

数据标注工具为 Anaconda 的 labelImg，本课题仅检测人、异物和鸟巢三大类别，其数据样本如下图所示，故标注信息为 human、entanglement、nest，具体对应信息如下表所示。标注步骤为选择标注路径、标注格式和标注信息存放路径，然后按 w 键后选中需要标注位置，输入对应的标签并回车，从而完成一次标注。对于 VOC 的标注格式，

会产生一个 xml 标注文件，paddleDetection 则是根据 xml 所包含的信息处理数据集。



图 4.1 标注数据样本

xml 文件中包含以下字段：filename 表示图像名称；size 表示图像尺寸，包括图像宽度、图像高度、图像深度；object 字段表示每个物体，包括 name（目标物体类别名称）、pose（关于目标物体姿态描述）、truncated（目标物体目标因为各种原因被截断）、occluded（目标物体是否被遮挡）、difficult（目标物体是否是很难识别）、bndbox（物体位置坐标）。

最终标注文件树如下：

```
├── annotations
│   ├── road0.xml
│   ├── road1.xml
│   ├── road10.xml
│   └── ...
├── images
│   ├── road0.jpg
│   ├── road1.jpg
│   ├── road2.jpg
│   └── ...
├── label_list.txt
├── train.txt
└── valid.txt
```

表 4.2 标注信息

类型	名字	数量(张)
人	human	1000
异物	entanglement	1000
鸟巢	nest	1000

其中 label_list.txt 是类别名称列表，指定了类别名称和类别 id 的映射关系，train.txt 是训练数据文件列表，每一行包括一条图像路径和图像对应标注 xml 文件路径，valid.txt 是验证数据文件列表。

4.2.2 训练&评估

标注完数据集之后，下一步便是在服务器上利用这些数据集进行训练、评估和预测。这些全部由 paddleDetection 完成，大致过程如下图所示。以路标检测为例进行说明。

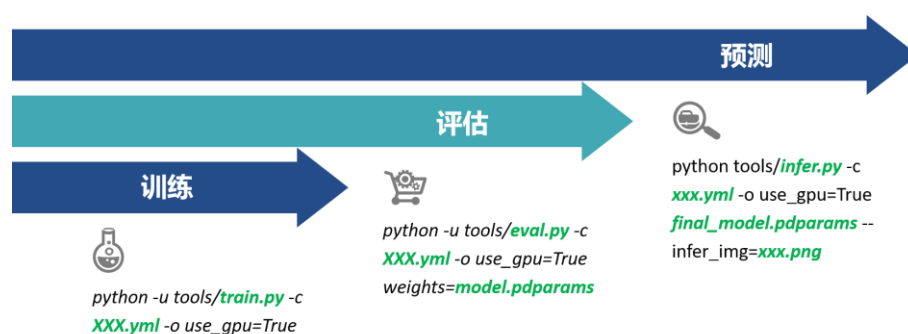


图 4.2 paddleDetection 处理过程

在开始训练之前，需要选择合适的 YOLOv3 模型文件，并且需要根据标注信息、检测目标和训练目的修改 yolov3_mobilenet_v1.yml 或者 yolov3_DarNetwork.yml 的相关配置参数，例如 max_iters（最大迭代次数）、metric（mAp 评估方式）、num_classes（数据集类别数）、backbone（框架）以及数据路径、模型输入、数据增强参数设置。

模型配置文件设置完毕后，开始移植训练代码（train.py）、和评估代码（eval.py）。train.py 主要用于读取 YOLOv3 配置文件，预加载训练模型配置信息，例如是否使用 GPU、YOLOv3 框架、训练参数设置命令等内容，配置好相关信息后调用 PaddlePaddle 的训练库进行训练，并最终由 Linux 终端命令进行调用，其训练相关设置命令如下表所示。另外，训练、评估和预测等终端命令的执行目录必须在 paddleDetection 对应目录下。

表 4.3 常用可选参数

FLAG	支持脚本	用途	默认值	备注
-c	ALL	指定配置文件	None	必选
-o	ALL	设置或更改配置文件里的参数内容	None	更高优先级
--eval	train	是否边训练边测试	False	
--slim_config	ALL	模型压缩策略配置文件	None	
--use_vdl	train/infer	是否使用 VisualDL	False	VisualDL 需 Python>=3.5
--output_eval	eval	评估阶段保存 json 路径	None	默认为当前路径
--output_dir	infer/export_model	预测后结果或导出模型保存路径	./output	
--infer_dir	infer	用于预测的图片文件夹路径	None	--infer_img、--infer_dir 必须至少设置一个
--infer_img	infer	用于预测的图片路径	None	infer_img 具有更高优先 级

```
PaddleDetection
├── configs # 提供所有模型、算法、扩展模块的配置脚本
├── contrib
├── dataset
├── demo
├── deploy # 提供C++端和python端使用飞桨预测库进行推理部署的详细指导
├── docs
├── ppdet #处理模块，加载数据并将其转换成适用于物体检测模型的训练、评估、推理所需要的格式
├── slim
├── tools # 一键式训练、评估、推理、模型导出`train.py` `eval.py` `infer.py` `export_model.py`
├── .gitignore
├── .pre-commit-config.yaml
├── .style.yapf
├── .travis.yml
├── LICENSE
├── README.md
├── README_en.md
└── requirements.txt # 列出了PaddleDetection的所有依赖库PaddleDetection
```

图 4.3 paddleDection 目录

标注数据集、修改 YOLOv3 配置文件和移植训练和评估程序代码之后，便可在 paddleDection 根目录下使用 Linux 终端命令调用 train.py 和 eval.py 进行评估和预测，输入“python -u tools/train.py -c configs/hw_configs/yolov3_.yaml -o use_gpu=True --eval”（命令参数见第 35 页表 4.3）便可同时进行模型的训练和评估（在训练中交替执行评估，评估在每个 epoch 训练结束后开始。每次评估后还会评出最佳 mAP 模型保存到 best_model 文件夹下）如果需要将过程可视化，可添加 visualDL 可视化工具，将 loss 曲线可视化，点击可视化按钮，设置 logdir 和模型文件，就可以对训练过程 loss 变化曲线和模型进行可视化，示意图如下，其他可视图如下表所示。训练完成后会训练产生模型文件 model.pdparams，与此同时评估之后会生成 mAp 值和 Loss 曲线，将训练模型可视化、数值化。

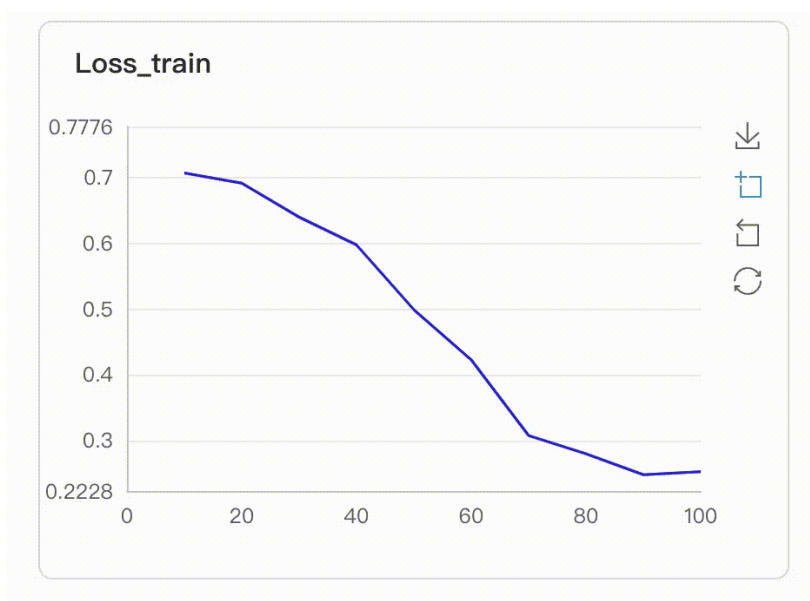


图 4.4 Loss 曲线

表 4.4 可视图表

组件名称	展示图表	作用
Scalar	折线图	动态展示损失函数值、准确率等标量数据
Image	图片可视化	显示图片，可显示输入图片和处理后的结果，便于查看中间过程的变化
Audio	音频播放	播放训练过程中的音频数据，监控语音识别与合成等任务的训练过程
Graph	网络结构	展示网络结构、节点属性及数据流向，辅助学习、优化网络结构
Histogram	直方图	展示训练过程中权重、梯度等张量的分布
PR Curve	折线图	权衡精度与召回率之间的平衡关系，便于选择最佳阈值
High Dimensional	数据降维	将高维数据映射到 2D/3D 空间来可视化嵌入，观察不同数据的相关性

4. 2. 3 预测

PC 端的预测是由 paddleDetection 中的 infer.py（预测代码）完成的，该部分仅用来测试模型训练效果并进行调试，调试之后又的模型还需要进行压缩、转化，部署到树莓派中运行。其移植内容与 train.py 和 eval.py 类似，paddleDetection 源码是以 PaddlePaddle 预测库（Python3 第三方库）为依赖，最终实现预测功能的。通过运行命令“python infer.py -c yolov3.yml -o use_gpu=True weights=model.pdparams --infer_img=xxx.png”（命令参数见第 35 页表 4.3）来实现，其中 weights 指名预测所用的权重路径（即训练评估得出的最终预测模型），预测前后的实例图片如下图所示。



图 4.5 预测效果

从准备数据集到配置模型文件，再经过训练、评估，再到最后的图片预测，整个过程是在 `paddleDetection` 和 `paddlepaddle` 的环境中进行，并以 YOLOv3 模型为基准。但由于最后的模型预测是需要在移动嵌入式端运行的，且树莓派并不支持 `paddleDetection` 和 `paddlepaddle`，所以需要通过 `paddleLite` 来进行预测，因此此刻的模型只能运行在 PC 端，若想部署到移动端，还需要导出模型并使用 `opt` 工具转化为 `paddleLite` 模型（.nb 文件），经过处理后的模型便可以在移动端使用了。

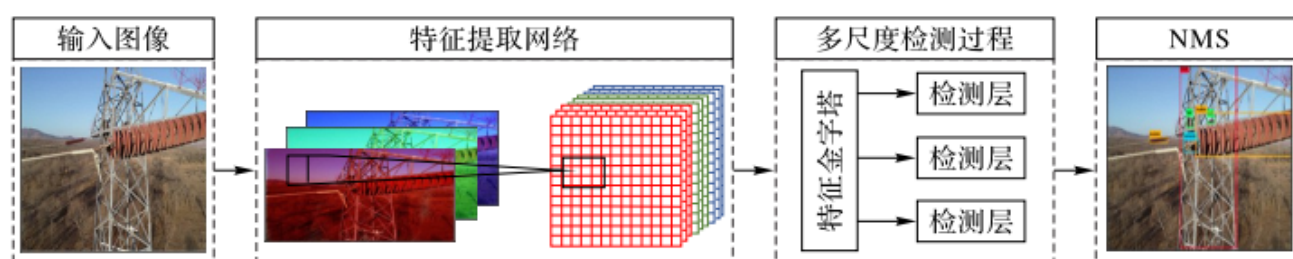


图 4.6 YOLOv3 预测过程

4.2.4 压缩

当 `paddleDetection` 原训练模型文件的复杂度和体积较大时，会消耗移动端过多内存以及降低运算速度，此时可使用基于 `paddleSlim` 的模型剪裁蒸馏联合训练或者剪裁量化联合策略压缩模型。首先用对应的 Slim 策略模型配置文件替换原来的 `yolov3` 配置文件，重新按照上述顺序进行训练、评估、预测，获取压缩之后的模型。因为本课题研究的高压线异常检测仅有人、鸟巢、缠绕物，且都是基于官方程序进行移植，模型文件体积不大，且可以直接导出转化在移动端运行，并且模型压缩策略仅支持 YOLOv3-MobileNet，故本课题不对训练模型进行压缩。

表 4.5 剪裁-Pascal VOC

模型	压缩策略	GFLOPs	模型体 积(MB)	输 入 尺 寸	预测时延 (SD855)	Box AP
YOLOv3-MobileNetV1	baseline	24.13	93	608	289.9ms	75.1
OLOv3-MobileNetV1	剪裁 -l1_norm(sensity)	15.78(-34.49%)	66(-29%)	608	-	78.4(+3.3)

4.2.5 模型导出与转化

因为需要将 `paddleDetection` 训练好的模型部署到移动嵌入式端的 `paddleLite` 中运

行，且 paddleLite 预测库采用 C++，所以需要通过 paddleDecton 中 tools 文件下的 export_model.py 将模型导出，导出模型的命令为“python export_model.py -c yolov3.yml -o weights=model_final.pdparams --output_dir ./inference, export_model.py”（命令参数见第 35 页表 4.3）导出的文件为__model__（模型）和__params__（参数），之后还需要通过 paddleLite 的 opt 转化模型将__model__和__params__转化为部署端预测模型 model.nb，进而部署到移动端的 paddleLite 当中。

4.3 模型部署

首先在树莓派端源码编译安装 paddle-Lite2.8，然后在根目录下执行 download_models_and_libs.sh（可执行脚本文件）获得最新的本地模型和预测库（armv8 架构的 libpaddle_light_api_shared 和相关 C++头文件），其目录结构如图所示：

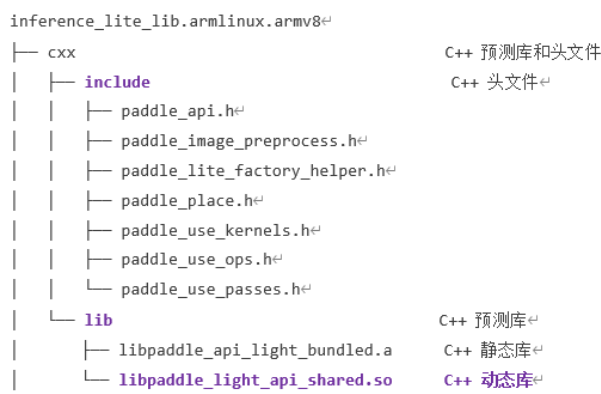


图 4.7 paddlelite 文件树

之后搭建 paddle Lite Demo 工程文件，其文件树如图所示：



图 4.8 paddlelitedemo 文件树

object_dection_demo.cc 为模型预测程序，CmakeLists.txt 用于设置编译信息，run.sh 为执行预测的脚本文件。

4.4 模型预测

paddleLite 预测共有 5 个步骤，分别为模型配置、创建预测器（Predictor）、设置输入数据、启动推理和获取输出结果。模型配置主要用来设置预测所使用的 CPU 线程数（CPU_THREAD_NUM），运行模式（CPU_POWER_MODE）和预测模型（xxx_model）；每一个模型对应一个预测器，设置的输入数据作为函数参数传入 Predictor 的 Process 函数；启动推理通过调用 OpenCV 相关库函数启动 paddleLite 的模型预测；然后获取最终的输出结果，并根据预测效果在视频流中实时绘制识别边框，并在终端打印输出被识别物体的位置和置信度，从而完成硬件端的模型预测。

4.5 关键技术

本部分的关键技术为 paddleLite 模型移动嵌入端部署和 mjpg-streamer 视频流传输。一般情况下，都是在远程服务器上对采集的视频进行预测，但是本课题以树莓派为硬件载体，如果按此方案实现，树莓派需要将采集的视频发送至远程服务器，也就是在 PC 端处理数据，但是这里涉及到问题就是视频传输问题，视频的网络传输本身就存在延迟，况且还需要边移动边进行预测，预测识别部分也需要一定的时间，因此增加了一定的难度，所以换成树莓派端进行预测识别，虽然树莓派不支持 paddleDection 和 paddlepaddle，但是可以安装百度飞桨的轻量化推理引擎 paddleLite，在该引擎上进行模型预测，基本不存在任何时延问题帧率稳定在 40 以上，而模型的训练、评估、导出和优化交给本地 PC 完成，这样二者的性能都能够充分发挥。

虽然在轻量化引擎上运行预测模型，但是还是需要将预测视频流传输到网页中，便于实际工作中的监测与查看，由于 MJPG 具有帧率高的优点，而 YUYV 虽然画质比 MJPG 高，但是帧率较低，实际应用中，视频流传输的重要性大于视频传输之后的画质，因此 mjpg-streamer 选用了 MJPG 分辨率格式，但是可以将 YUYV 转换为 MJPG 进行视频传输，它主要是通过 JavaScript 编写的网页程序和本地视频处理代码来实现视频的高速传输，最后通过同一个局域网的 IP 地址加端口号访问本地视频页面。

最关键的部分是将 paddle-Lite 嵌入到 mjpg-streamer 中，使其在远程网页中可以查看实时预测视频流。由于 mjpg-streamer 提供了 opencv 接口，因此主要步骤为修改

Makefile 文件，使其 mjpg-streamer 在编译安装后产生 input_opencv.so，之后通过 input_opencv.so 和 OpenCV 库获取原始视频数据并输入到处理程序当中，然后再通过 output_http.so 将视频流传输到网页当中，其中需要修改 input_opencv 文件中的 C++版本的过滤器，内容为 PC 端训练转化后的模型，预测脚本和 C++预测程序，过滤器由 input_opencv.so 调用，也就是说主要将 paddleLite 工程作为过滤器放入到 mjpg-streamer 的 filters 中，这样 mjpg-streamer 在启动之后会自动调用预测程序进行预测，并将带有预测的视频传输至网页当中，从而实现高压线异常巡检的远程实时监测。

4.6 本章小结

本章按照自上而下的顺序介绍了预测模型是如何产生的，又是如何部署到树莓派端的，以及部署之后的预测结果又是如何传送至网页端进行实时监测，整个过程如图所示。

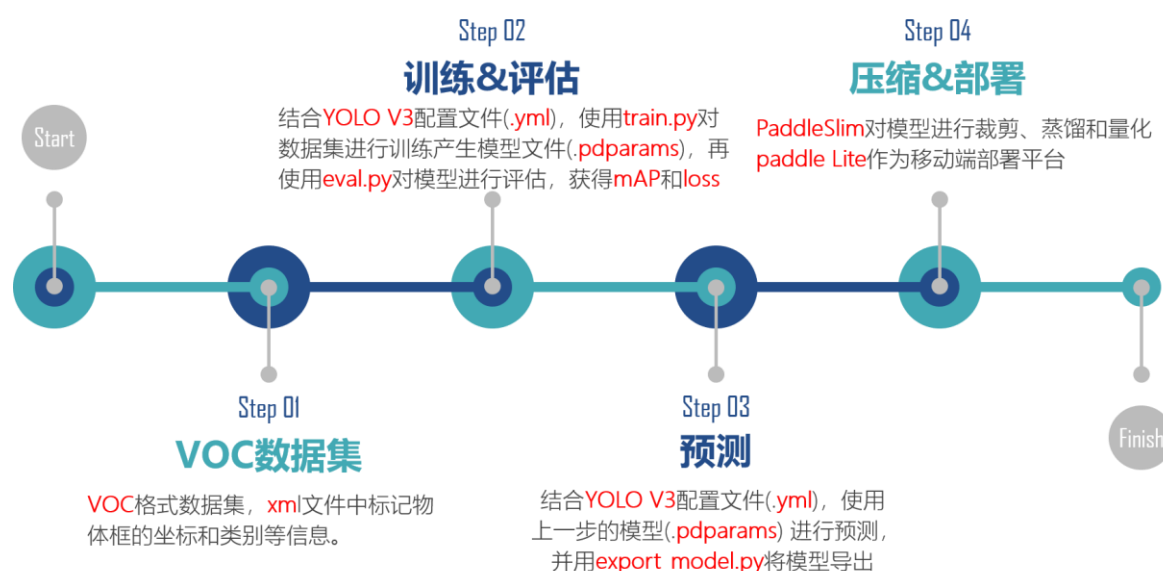


图 4.9 模型预测过程

5 实验过程

本章主要叙述整个高压线巡检机器人研究过程中的实际内容和产物，依旧按照自下而上的顺序阐释底层搭载平台的制作和控制代码的编写，以及应用层目标检测的实现和视频流的传输。

5.1 机器人实现

5.1.1 结构制作

制作材料第二章已经说明，此处不再赘述。首先利用亚克力板、直流电机、轮胎制作四轮驱动小车的主题架构；然后安装移动电源、锂电池、L298N 模块，并制作电源线、转接板，电源开关，从而完成电源模块的线路布置；之后固定二自由度云台、超声波云台、超声波、RGB 灯、USB 摄像头、树莓派摄像头，并完成它们之间的线路布置，具体线路见附录 B；之后对线路进行整理、捆绑和固定，然后安装轮胎防护罩和科大校徽。至此，完成机器人搭载平台的制作，其外观如下。

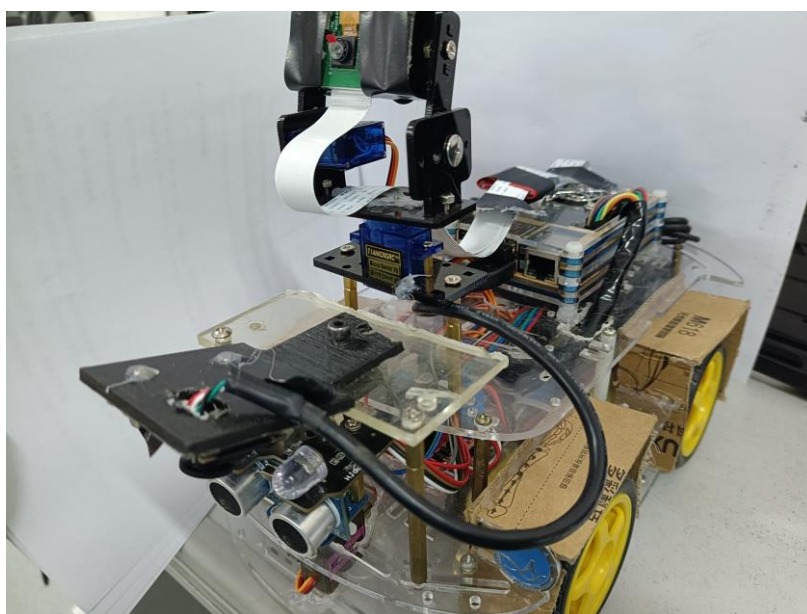


图 5.1 机器人外观图

5.1.2 控制编写

控制程序的编写环境为树莓派 Debian 64 位系统+Python3.7.3，编写 IDE 为 Thonny，编辑页面如下图所示，主要使用的第三方库为 pygame、traceback、threading 和 RPI.GPIO。最终效果为无线键盘控制四轮驱动小车底盘的前进、后退、转弯、舵机转向、RGB 提示灯。

5.2 模型制作

5.2.1 数据集标注

标注工具为 Anaconda3 的 labelImg，环境为 python3.7，标注格式为 PascalVOC，标注系统为 Win10。标注步骤如下图所示：

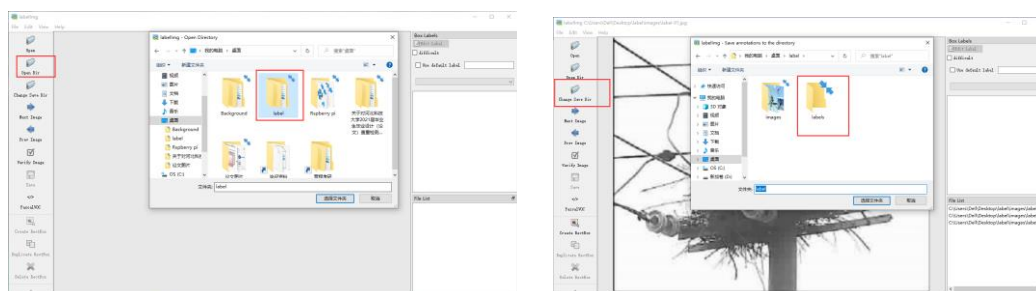


图 5.2 选择标注路径



图 5.3 选择标注格式 VOC

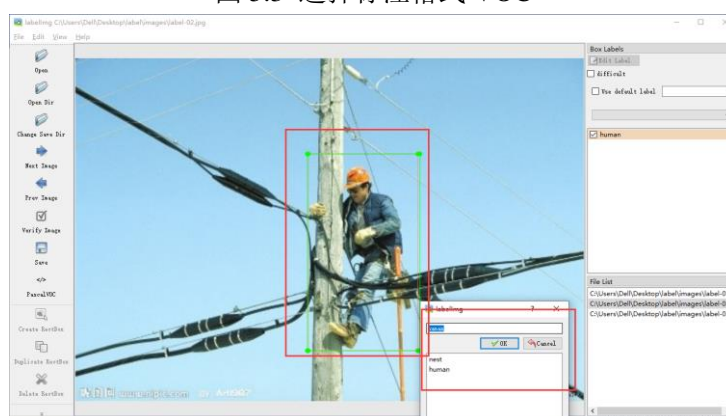


图 5.4 进行标注

5.2.2 模型训练

使用百度 AI 平台的服务器进行训练，编程在使用本地的 VS Code，主要编写 PC 端的 train.py、eval.py、infer.py、export_model.py 和配置文件 yolov3_darknet.yml。

训练、评估、预测命令如下（命令参数见第 35 页表 4.3）：

“python -u tools/train.py -c configs/hw_configs/yolov3_darknet.yml -o use_gpu=True”

```
“python -u tools/eval.py -c configs/hw_configs/yolov3_darknet.yml -o use_gpu=True  
weights= yolov3_best_model.pdparams”
```

```
“python tools/infer.py -c configs/hw_configs/yolov3_darknet.yml -o use_gpu=True  
weights= yolov3_best_model_roadsign.pdparams  
--infer_img=dataset/voc/images/nest1.png”
```

获取的 loss 曲线和 mAp 值如图所示：

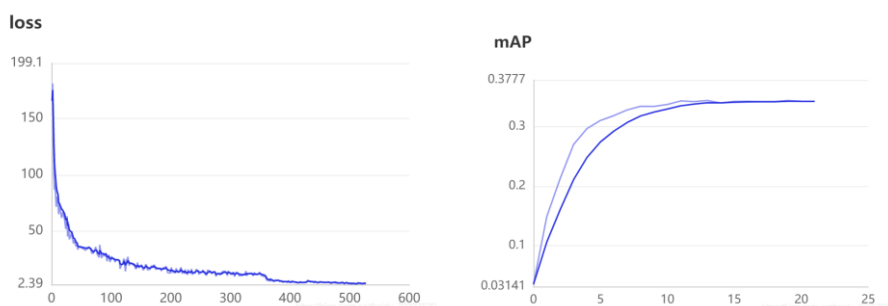


图 5.5 loss、mAp 曲线

PC 端运行预测模型前后的识别情况如下图所示：

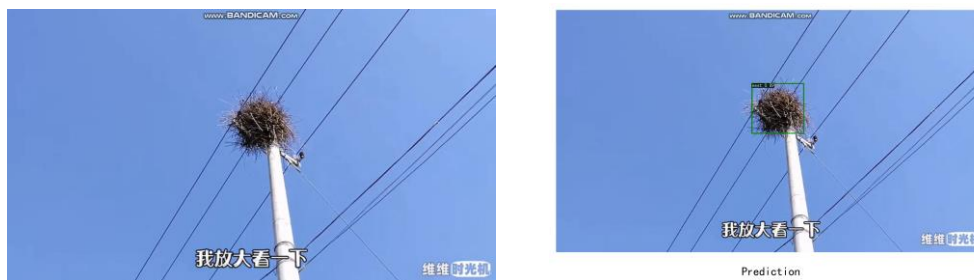


图 5.6 预测前后

5.2.3 模型转化

经过训练、评估后产生的权重，首先需要 `export_model.py` 导出模型，会生成两个文件，分别为__model__（模型）和__params__（参数），然后再使用 `paddleLite` 的 `opt` 转换模型，将__model__和__params__转化为可以运行在 `armv8` 架构的模型 `model.nb`。

5.3 移动端预测

5.3.1 paddleLite 部署

树莓派端源码编译安装 `paddleLite` 的命令如下（参数解释见第 45 页表 5.1）：

```
git clone https://github.com/PaddlePaddle/Paddle-Lite.git
```

```
cd Paddle-Lite && git checkout release/v2.8
```

```
rm -rf third-party
```

```
export LITE_BUILD_THREADS=4

./lite/tools/build_linux.sh --with_strip=ON --opt_model_dir=PaddleLite/opt_model
```

编译之后会在 Paddle-Lite/build.lite.linux.armv8.gcc/inference_lite_lib.armlinux.armv8 中生成最新的模型和预测库，安装之后进入 python3 环境，import paddlelite 验证安装，最后把生成文件添加至 paddleLite 工程文件中的 PaddleLite 文件夹中。

表 5.1 相关命令参数

命令参数	含义
sh	执行脚本
xxx.sh	脚本文件
git clone	克隆 GitHub 远程仓库
cd	打开文件
git checkout	切换分支
rm -rf	删除文件
third-party	paddleLite 预测库下载文件
export	环境变量设置
--with_strip: (OFF ON)	是否根据输入模型裁剪预测库，默认为 OFF
--opt_model_dir	输入模型的绝对路径，需要为 opt 转化之后的模型

5.3.2 预测程序

依据官网 paddleLiteDemo 创建预测工程，其中 code 文件夹为预测程序，paddleLite 文件夹为最新模型和预测库。

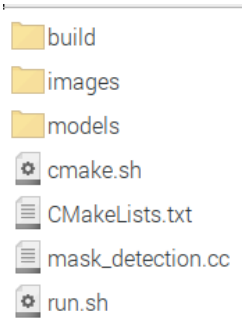


图 5.7 预测目录

5.3.3 paddle-Lite 嵌入

若想远程监控预测画面，还需要将 paddleLite 嵌入到 mjpg-streamer 中，并结合 OpenCV3.2.0 实现。嵌入方式主要是把预测相关文件放

在 `./input_opencv/filters/cvfilter_cpp` 文件中，并将 C++ 预测程序内容写入 `cvfilter_cpp` 下的 `filter_cpp.cpp` 中，注意只能更改内容，然后将 PaddleLite 下的模型和预测库发到 `mjpg-streamer-experimental` 中，由 `cvfilter_cpp` 下的配置文件 `CMakeLists.txt` 调用，训练模型由 `filter_cpp.cpp` 调用，预测相关目录和 `mjpg-streamer` 根目录如下图所示。

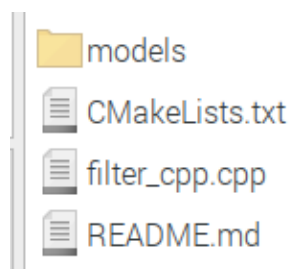


图 5.8 Filter 目录

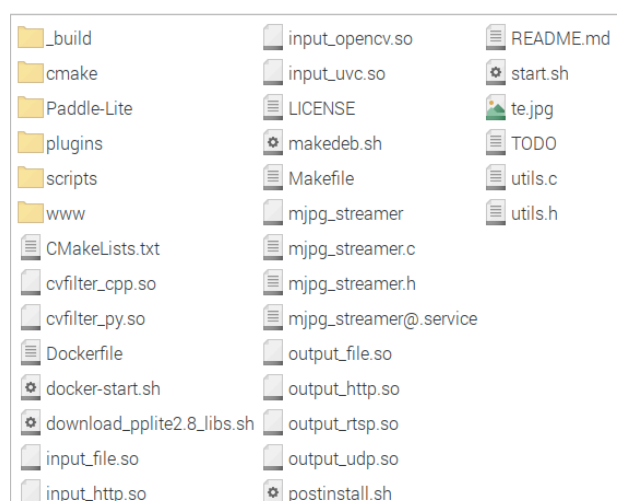


图 5.9 mjpg-streamer-experimental 目录

之后便可以开启预测，并且可以在网页中进行查看，执行命令为“`sh download_pplite2.8_libs.sh && make && sh start.sh`”然后通过局域网设备使用浏览器访问树莓派 ip 的 8080 端口，即可查看到一个实时检测的画面。

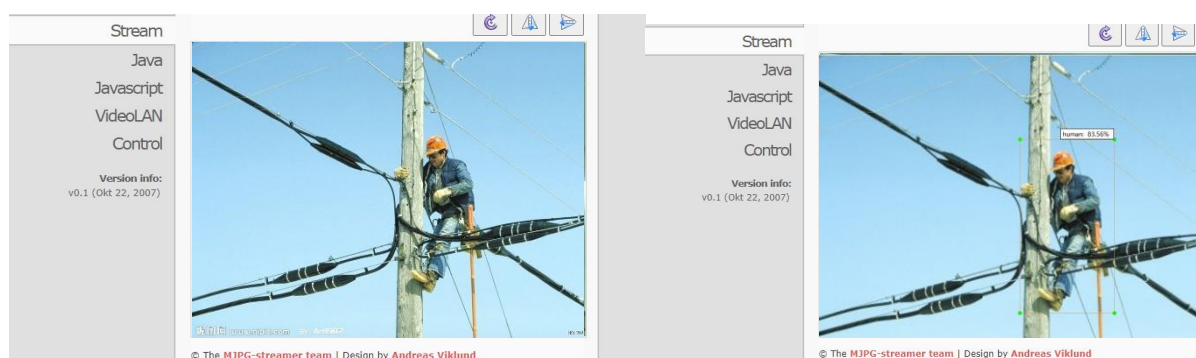


图 5.10 mask

5.4 本章小结

本章详细介绍了每个功能模块的架构、内容和实现步骤。先是机器人的结构实现和控制程序的编写和调试，然后是 PC 端模型训练、导出、转化的步骤、命令和注意事项，最后是树莓派端 paddle-Lite、OpenCV 和 mjpg-streamer 的源码编译安装以及 paddle-Lite 嵌入操作，最终实现硬件端便移动便检测，软件端可通过网页访问摄像头数据，可以说是一个完整的物联网产品从构思到设计到实现到优化到测试完善的明确体现。

结 论

本课题主要分为两大部分，分别为软件端和硬件端。硬件端为高压巡线机器人搭载平台，主要用于搭载巡检摄像头并实现基本的运动姿态以满足工作人员和工作场地的需求，因此实现了自动避障、自动巡线、键盘操控机器人等核心功能，同时考虑到工业中的实际应用型，对线路、外观进行了美化，对关键部件进行了保护，虽然本课题中的四轮驱动小车为工业中巡检装甲车的“阉割版”，但是“麻雀虽小，五脏俱全”，完全可以初步实现巡检装甲车的核心功能。从实际上来讲，硬件端最重要的是电路设计和程序编写以及模块的可移植性，只有这样的设计才能够适合多数硬件开发平台，而不是把精力耗费在开发平台的选型和设计上面。虽说结构设计不重要，但也要和控制程序与控制电路结合起来，真正使三者融为一体。软件端又分为两个部分，分别为 Server 和 Raspberry 端。Server 端借助于使用 paddlepaddle 等深度机器学习技术框架和 YOLOv3 目标具体检测组件算法，在此基础上通过整合 paddleDetection 等的开发检测组件功能来直接投入实现对一个目标具体检测模型的系统训练、评估、预测、导出和训练转化，该检测模型进行训练转化过程中最重要的环节便是构建目标化的检测模型，一般的实际情况下，都是需要使用 GPU 进行训练，主要因为 GPU 的处理能力远远超过 CPU，框架深度学习就是利用庞大的数据集来对计算机进行指导，此过程非常复杂并且消耗电脑的 CPU、GPU、显存、缓存和内存，因此训练模型的最佳环境便是远程服务器而不是本地笔记本电脑。另一个部分便是 Raspberry 端，树莓派主要用于运行预测模型，它的 CPU 和 GPU 处理能力并不足以支持它进行深度学习，并且效率低下，但是使用轻量化引擎 paddle Lite 进行模型的预测却绰绰有余，况且可以避免视频在树莓派和 Server 端来回传输并识别的延时问题，但是同时也需要在树莓派端将预测视频流传输到网页当中进行远程实时监控，此处便需要将 paddleLite 加以训练模型共同嵌入到 mjpg-streamer 当中，并借助 OpenCV3 来实现此功能。这样一来，便真正的实现了软硬结合。

硬件端设计技术的难点在于自动行驶部分，需要设计严谨的行为逻辑，然后通过 USB 摄像头和超声波测距模块来实现四轮驱动小车的自动巡线和自动避障，其实该车的

实际应用中还需要添加激光雷达实现全场定位，这样的话，才可以提升自动行驶的精确度。而自动巡线的基本原理又是 OpneCV 的边缘检测，目前采用的算法只是比较基础的实现方法，其实目前已经有多优化的边缘检测算法，既能提高精确度，又可以提升程序运行速度。总而言之，激光雷达的添加和边缘检测算法的优化算是本课题硬件端的不足之处。

软件端设计的技术难点在于视频传输延迟和模型训练精确度。首先说延迟问题，虽说 Server 端训练模型，树莓派端跑模型的方式可以解决视频来回传输的时延问题，但是随之而来的是树莓派单方视频传输时延问题，目前本课题只能通过增加给分配目标检测的 CPU 数量来降低延迟，但是树莓派仅有 4 个 CPU，允许同时 4 个进程运行，同时还有分配一个进程给机器人的控制系统，所以该问题目前仍旧亟待解决。再说模型的训练精度问题，一方面是网上没有关于高压线的开源数据集，所以我们采集标注的数据集有限，数量很少；另一方面是知识虽广但不深入，在使用 PaddlePaddle 的过程中，大多数功能都是通过移指代码和修改参数实现，并不了解每一个函数方法的底层实现，也就是说只能使用，自己并无法去优化目标检测算法，仅能通过提升硬件配置来缓解此问题，总而言之，网络传输延迟和模型训练算法优化是本课题需要继续改进提升的地方。

只有经历过一个完整项目的设计、开发、调试、完善、优化迭代的全部过程，才能够体会到社会工作的不易与艰难，当然更多的是产品研发的技术和技巧，其中更重要的产品研发技巧，因为一个好的产品研发方案，和一个好的产品技术路线可以造成事半功倍的效果，在此之前还需要做好充分的市场调查和技术调查，从而确定正确可行的技术方案，而不是盲目的选择自己熟练技术和方案，很有可能最后功亏一篑。

参考文献

- [1] 周自更, 王胜伟, 赵小萌, 等. 高压输电线路巡线机器人的研究与设计[J]. 电气时代, 2020(06): 80-82.
- [2] 董浩. 基于高压输电线路的智能巡线机器人[J]. 电子世界, 2019(11): 158-159.
- [3] 赵钰佳. 高压线巡检机器人系统设计及其检测技术研究[D]. 哈尔滨工业大学, 2020.
- [4] 李威. 高压输电线路巡线机器人结构设计[D]. 华北电力大学(北京), 2019.
- [5] 王恒涛. 基于 TensorFlow、Keras 与 OpenCV 的图像识别集成系统[J]. 电子测试, 2020(24): 53-54+124.
- [6] 谢永康. 深度学习开发平台与应用[J]. 人工智能, 2020(03): 43-53.
- [7] 吴雪, 宋晓茹, 高嵩, 等. 基于深度学习的目标检测算法综述[J]. 传感器与微系统, 2021, 40(02): 4-7+18.
- [8] 翁智, 程曦, 郑志强. 基于改进 YOLOv3 的高压输电线路关键部件检测方法[J]. 计算机应用, 2020, 40(S2): 183-187.
- [9] 马艳军, 于佃海, 吴甜, 等. 飞桨: 源于产业实践的开源深度学习平台[J]. 数据与计算发展前沿, 2019, 1(05): 105-115.
- [10] 郭振宇, 高国飞. 基于 YOLO v4 的复杂路口下人车混行检测算法研究[J]. 信息技术与信息化, 2021(02): 236-240.
- [11] 王雅轩, 高露, 朱明佳, 等. 基于 Tiny-Yolov3 模型的 4 位数码管模式识别[J]. 电脑编程技巧与维护, 2020(12): 122-123+161.
- [12] 葛雯, 王嘉利. YOLO-K 模型多目标检测算法研究[J]. 电脑与信息技术, 2021, 29(02): 27-30.
- [13] 郭磊, 吴功平, 麦晓明, 等. 高压输电线路巡检机器人续航里程的预测方法[J]. 武汉大学学报(工学版), 2018, 51(04): 347-355.
- [14] 杨磊. 高压输电线路巡检数据库及其管理系统[J]. 科技风, 2018(10): 130.

[15] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. arXiv, 2018.

致 谢

四年时光白驹过隙，就如同做梦一般飞快，蓦然回首，入学竟在灯火阑珊处，不知不觉，我大学时光就在实验室、自习室、餐厅中度过了，心中无怨无悔，无仇无恨，到最后收获的是计算机专业相关知识、较强的实践能力以及西安电子科技大学的入学通知书，又可以继续在嵌入式方面继续深造，为国争光！记得从大一入学申请入党、加入实验室、努力学习公共课知识到大二加入机器人队，参加互联网+、三创赛、挑战杯、学科竞赛，再到大三的考研复习，大四的毕业设计，每一个阶段自己都没有虚度，更重要的是，都遇到了自己生命中不可或缺德贵人和恩人，没有他们，就没有今天的我，俗话说的好，人要常怀一颗感恩之心，只有这样才能走得更远、走得更高、学的越深！因此，此时此刻我要向我的贵人们发表致谢。

首先是我的毕设指导老师许云峰，虽然每次开会都很严肃，但是对我们每一个人都很上心，无论是在文档方面、编程方面，甚至为人处世方面都给予了我莫大的启示，他是我学术道路上的启明星，是我茫茫人海的指路人，是学习工作的领头羊，他让我明白了什么叫做真正的搞学术，在整个毕设过程中也帮助我解决了很多问题，在此对老师说一声感谢！其次是大学四年一直不断指导、帮助我的辅导员老师李俐莹，是她给予了我进步的机会，给予了我表现的机会，给予了我大学的希望和光，是她带领我走上了进步的阶梯；是她给予了我精神上的知识。对于个人、班级和学院而言，她称得上是一位好老师；最后是一直在辅导我帮助的我的舍友和同学们，他们是胡博谭、张贺俊、张建成、付贵东、侯月岭、刘玉航和李竹林，没有他们，可能如今的论文和作品并不完美，在此表示深深的感谢。

除此之外，我还要感谢指导我参加三创赛的王晓红老师，大三期间，帮助王老师一起完成养老机器人技术文档的编写并在比赛中取得了优异的成绩，那是我第一次接触深度学习，是王晓红老师给予了我深度学习的萌芽，没有王晓红老师，可能我的题目并不会和深度学习相关。然后是大一给予我大学数学启蒙的数学老师刘萍，没有她，我的数学可能并不会得到提升。

再者，我感谢一直和我一起参加科技竞赛，共同考研复习的两位挚友，他们分别是张成和曹晓东，没有他们，就没有如今的科研成绩。在科研道路上，我的两位挚友一直在指导我，帮助我，陪伴我。两年来，我们各自都取得了优异的成绩，非常感谢大三、大四有他们的陪伴。

我最感激的是生我养我的父亲和母亲，作为一个贫苦农村出生的大学生，深知父母日夜操劳、含辛茹苦抚养我长大成人的不易与心酸，深知农村学生进入大学的困难与无助。然而，正是他们在我慢慢人生路中不断给予我精神和物质上的支持，他们是世界上我最信赖最亲近的人。

此时此刻，我的脑中还有许多想要感谢的人，但我的感激之情已经无法用言语表达，那么就以“感谢”二字来结束我的毕业论文吧。

附录 A

表 A.1 键值对应表

键值	动作	前提模式
K_ESCAPE	退出系统	Manual/Automatic
K_SPACE	开启控制	Wait
K_END	关闭系统	All
K_HOME	重启系统	All
K_w	前进	Manual
K_s	后退	
K_a	左转	
K_d	右转	
K_a+K_LCTRL	原地左转	
K_d+K_LCTRL	原地右转	Manual/Automatic
K_KP0	云台归中	
K_UP	向上转动	
K_DOWN	向下转动	
K_LEFT	向左转动	
K_RIGHT	向右转动	
K_KP5	超声波舵机归中	
K_KP4	向左转动	
K_KP6	向右转动	
K_KP_MINUS	超声波测距	
K_1	手动模式	Manual/Automatic
K_2	自动避障模式	Automatic-Avoidance
K_3	自动巡线模式	Automatic-Trackline

附录 B

表 B.1 GPIO 引脚对应表

BOARD	BCM	pin	Module
27	16	ENA	Motor
28	20	IN1	
29	21	IN2	
24	19	IN3	
25	26	IN4	
23	13	ENB	Gimbal
2	27	YAW	
0	17	PITCH	
1	18	Red	RGB
4	23	Green	
5	24	Blue	
3	22	YAW	HC-SR04
21	6	Trig	
22	5	Echo	