



项目申请书

项目名称: 基于 OrangePi AIpro 和鸿蒙系统的智能网联小车开发

项目申请人: 戴鹏

联系方式: enactus_daipeng@163.com

申请日期: 2025 年 05 月 26 日

一、项目背景

1. 项目需求

随着无线通信和人工智能技术的快速发展，利用国产开发板及支持鸿蒙系统的移动终端开发智能网联小车，成为了开发者快速利用国产化软硬件技术掌握车联网开发的一种有效手段。本项目使用 OrangePi AIpro 开发板和高清摄像头搭建智能网联小车，将为智能小车的研究和开发提供重要的实践平台。

本项目内容（任务）主要分为以下三个部分：

- (1) 在 OrangePi AIpro 开发板上部署 YOLO 模型，通过摄像头辅助远程控制者进行目标检测，并在视频画面中显示目标检测结果。
- (2) 使用边缘检测算法提取每一帧图像的边缘信息，实现车道线检测、车道偏移计算等辅助控制功能。
- (3) 基于具有 wifi 通信功能的鸿蒙终端，开发应用程序实现控制者通过 wifi 远程控制智能网联小车运动，并将小车采集的视频数据能实时通过 wifi 传输到鸿蒙终端。

2. 技术方法及可行性分析

1. YOLO 算法

YOLO 是一种基于深度学习的单阶段目标检测算法，以其高速、高效和端到端的特性成为计算机视觉领域的革命性技术。该算法通过神经网络直接预测图像中目标的边界框和类别，适用于实时目标检测的场景，该算法的核心思想就是把目标检测转变为一个回归问题，利用整张图片作为网络的输入，通过神经网络，得到边界框的位置及其所属的类别。

以下是 YOLO 算法的具体实现步骤：

- (1) 划分图像

YOLO 将输入图像划分为一个固定大小的网格。

- (2) 预测边界框和类别

YOLO 算法预测出固定数量（通常为 5 个或 3 个）的边界框，每个边界框由 5 个主要属性描述：边界框的位置（中心坐标和宽高）和边界框包含的目标的置信度，以及预测目标的分类结果。

(3) 单次前向传递

YOLO 通过一个卷积神经网络（CNN）进行单次前向传递，同时预测所有边界框的位置和类别。相比于其他目标检测算法，YOLO 算法只需要一次前向传递即可完成预测，因此具有更快的速度，

(4) 损失函数

YOLO 使用多任务损失函数来训练网络，该损失函数包括位置损失、置信度损失和类别损失。其中位置损失衡量预测边界框和真实边界框之间的位置差异。置信度损失衡量边界框是否正确地预测了目标，并惩罚背景框的置信度。类别损失衡量目标类别的预测准确性。

(5) 非最大抑制

在预测的边界框中，可能存在多个相互重叠的框，代表同一个目标。为了消除冗余的边界框，YOLO 使用非最大抑制算法，根据置信度和重叠程度筛选出最佳的边界框。

本文给出以 YOLOv3 目标检测模型为例，给出模型结构如下图所示：

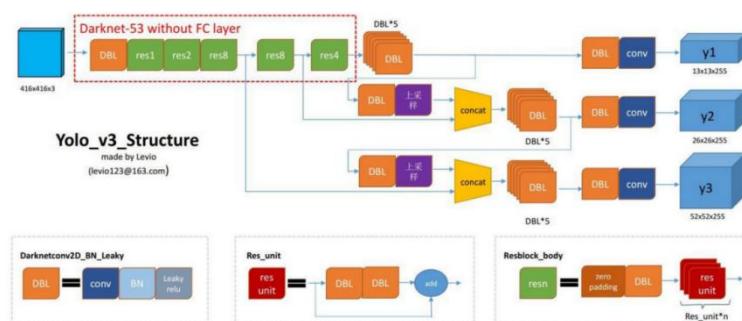


图 1 模型结构图

YOLO 算法的目标检测的速度非常快，标准版本的 YOLO 可以每秒处理 45 张图像，YOLO 的极速版本每秒可以处理 150 帧图像。这就意味着 YOLO 可以小于 25 毫秒延迟，实时地处理视频。YOLO 算法的迁移能力也比较强，可以快速运用到其他新的领域，完成特定需求的目标识别任务。

基于以上的特点，可以将训练好的 YOLO 算法模型部署至边缘计算设备，在适用于所要实现的目标识别任务同时，还保证识别的高实时性和高准确性。

2. 边缘检测算法

边缘检测算法是图像处理中用于识别图像中物体边界的关键技术，主要包括 Canny、Sobel、Prewitt、Laplacian 等经典方法。其中，因为 Canny 算子低错误率、

高定位精度和双阈值处理等特点，Canny 算子被认为是最优且最常用的边缘检测算。因此，车道线检测功能的实现，可以使用这种计算机视觉领域中常用的边缘检测算法——Canny 边缘检测算法。该算法的主要特点是能够通过提取每一帧图像的边缘信息，准确地检测到图像中的边缘，以突出最可能的车道线轮廓，同时对于噪声有较好的抵抗能力，可以过滤掉边缘检测图像中非车道部分的边缘信息，从而集中检测车辆前方的车道线。

以下是 Canny 边缘检测算法的具体构成：

(1) 灰度化处理

将输入的彩色图像转换为灰度图像。

(2) 高斯滤波

使用高斯滤波器对灰度图像进行平滑处理，以减少噪声的影响。

(3) 计算图像梯度

使用 Sobel 算子计算灰度图像在水平和垂直方向的梯度值，从而得到每个像素点的梯度强度和方向。

(4) 非极大值抑制

对于每个像素点，比较其梯度方向上的邻近两个像素点之梯度值的大小，只保留梯度方向上梯度值极大的像素点，有利于消除边缘上的模糊效果。

(5) 双阈值处理

将像素点分为三类：强边缘、弱边缘和非边缘。当像素点的梯度值高于较高的阈值时，将其分类为强边缘。当像素点的梯度值低于较低的阈值时，将其分类为非边缘。当像素点的梯度值介于较高和较低的阈值之间时，将其分类为弱边缘。

(6) 边缘连接

将弱边缘和其周围的强边缘连接起来形成完整的边缘。

以下是对一张图片应用 Canny 边缘检测算法使用的效果图：



图 2 边缘检测算法效果图

Canny 边缘检测算法具有高精度、噪声抑制和边缘连通性等优点，适用于本次智能网联小车的高质量边缘检测任务。在实际的应用中，还需要根据具体情况调整参数和优化算法，以达到最佳的车道线检测的效果。

3. 华为 ArkTS 语言和鸿蒙系统

(1) ArkTS 编程开发语言

ArkTS 是鸿蒙生态应用的开发语言，使用.ets 作为 ArkTS 语言源码文件后缀。提供了声明式 UI、状态管理等相应的能力，让开发者以更简洁、更自然的方式开发高性能应用。ArkTS 的重要特性之一是静态类型。相比于 TS 只在编译时进行类型检查，ArkTS 将编译时所确定的类型应用到运行性能优化中。由于在编译时就可以确定对象布局，对象属性的访问可以更加高效。面向未来，ArkTS 会结合应用开发以及运行时的需求持续演进，引入包括并行和并发能力增强、类型系统增强等方面的语言特性，进一步提升 ArkTS 应用的开发和运行体验。

(2) 鸿蒙系统

HarmonyOS 整体遵从分层设计，从下向上依次为：内核层、系统服务层、框架层和应用层。系统功能按照“系统 > 子系统 > 功能/模块”逐级展开，在多设备部署场景下，支持根据实际需求裁剪某些非必要的子系统或功能/模块。

HarmonyOS 技术架构如下图所示：



图 3 HarmonyOS 技术架构图

(3) 基于鸿蒙的 UI 界面开发

本次项目使用华为推出的 DevEco Studio 开发工具，通过 ArkTS 语言编写上位机用户控制界面，使得支持 HarmonyOS 系统的手机或平板电脑上的应用开发更加简单快捷。目前根据本次开源项目的基本要求，初步使用 DevEco Studio 开发工具设计了基于 ArkTS 语言的上位机界面，如下图所示：

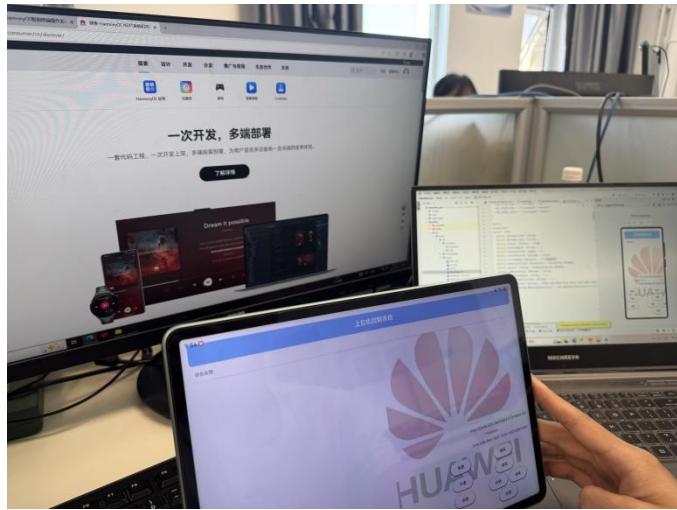


图 4 鸿蒙开发界面图

4. 开发板 OrangePi Alpro

（1）开发板介绍

OrangePi Alpro 是香橙派官方跟华为昇腾合作的新一代边缘计算产品，其使用华为昇腾 AI 技术路线，搭配 4 核 64 位处理器+AI 处理器，集成图形处理器，支持 8-12TOPS AI 算力。开发板还拥有相当丰富的接口，包括两个 HDMI 输出、GPIO 接口、Type-C 电源接口、支持 SATA/NVMe SSD 2280 的 M.2 插槽、TF 插槽、千兆网口、两个 USB3.0、一个 USB Type-C 3.0、一个 Micro USB（串口打印调试功能）、两个 MIPI 摄像头、一个 MIPI 屏等，并预留电池接口。

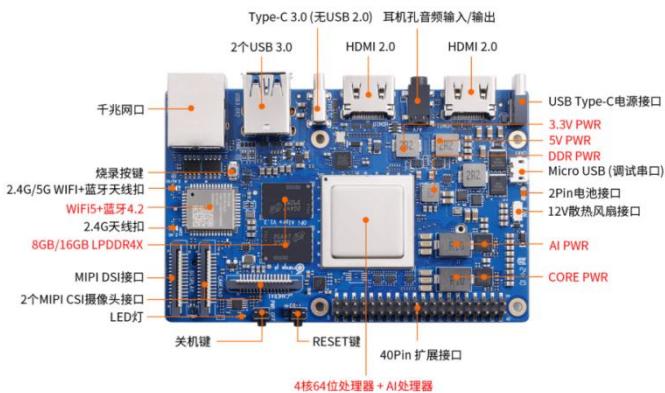


图 5 OrangePi AIpro 构造图

与传统的计算平台相比，OrangePi AIpro 具有低功耗和低成本的优势，其开放的软件生态系统可以支持多种主流的深度学习框架：MindSpore、TensorFlow 和 PyTorch，开发者可以利用这些工具进行开发和优化，快速实现自己的应用需求。

求。



图 6 OrangePi Alpro 内部图

(2) 开箱使用

OrangePi AIpro 的系统烧录完成后，将准备好的 HDMI 线和显示屏与 OrangePi AIpro 连接好相应的的接口，插上电源，即可进入开发板界面：



图 7 OrangePi Alpro 界面图

进入系统的 samples 目录，可以看到自带 8 个实验 AI 应用样例和一个启动 Shell 脚本：

```
(base) HwHiAiUser@orangepeiaipro:~/samples$ ls  
01-SSD 02-CNNCTC 03-ResNet50 04-HDR 05-CycleGAN 06-Shufflenet 07-FCN 08-Pix2Pix start_notebook.sh  
(base) HwHiAiUser@orangepeiaipro:~/samples$
```

图 8 OrangePi Alpro 案例图

通过启动 `start_notebook.sh` 来启动 Jupyter Lab:

图 9 Jupyter Lab 启动图

在 Jupyter Lab 中运行案例 3 ResNet50 分类，可得推理结果，初步验证开发板的 AI 推理能力：

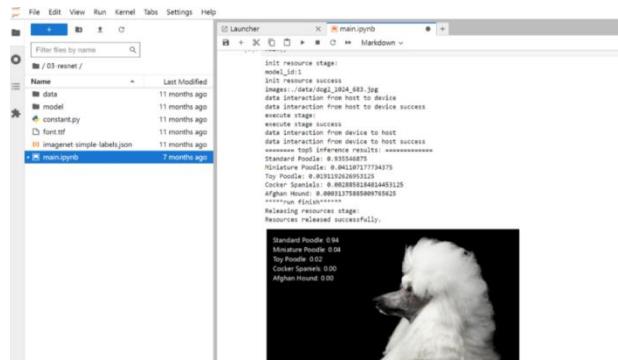


图 10 案例运行启动图

二、开发者本人具备能力和开源项目经验

1. 开发者本人具备能力

- (1).本人具有较为丰富的嵌入式硬件开发经验，熟悉 OrangePi Aipro 开发板的使用方法。
 - (2).熟悉 Linux 操作系统，可以熟练地使用 VNC、SSH 等工具进行远程开发调试。
 - (3).对 git 工具的使用比较熟悉，能够进行代码的版本控制和代码仓的管理。
 - (4).对于深度学习有较为丰富的学习和应用经验，可以熟练地使用华为云 ModelArts 等 AI 开发工具。

2. 开源项目参与和开发经验

(1) 课程参与经历

参与全国高等学校计算机类课程能力提升高级研修班（成都站），修习计算机昇腾课程：《基于昇腾 CANN 的 Ascend C 算子开发与应用开发实践》，全面系统学习基于昇腾 CANN 原生接口的高性能推理实践、基于 Ascend C 的算子开发应用实践、基于昇腾 CANN 科研实践等。

在培训设置有香橙派开发板实验环节，在参培过程中在老师的指导下进行了上手实操，了解了香橙派开发板的具体使用方法。以下是本次活动参与的结业证明：

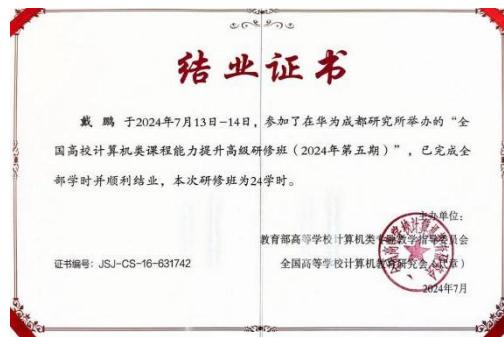


图 11 结业证书图

参与华为开发者学堂课程，顺利完成学习并通过了考核，获得 HarmonyOS 应用开发者基础认证证书，证书如下：



(2) 基于 OrangePi Aipro 开源项目经历

本人作为开发者参与华为 ICT 大赛中国教学赛，大赛最终晋级中国总决赛并获得全国三等奖。在赛项的案例中设计并开发了基于深度学习和 OrangePi Aipro 的调制类型盲识别系统。以下是实验的整体流程：

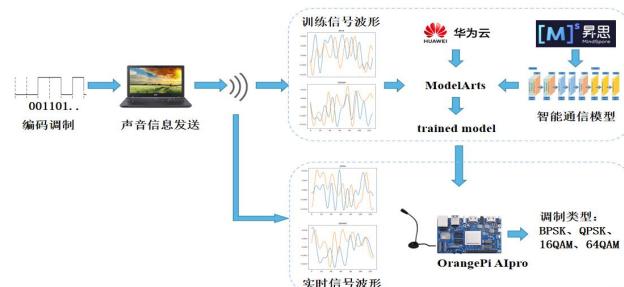


图 12 实验流程图

根据 AI 应用的全流程开发的顺序，使用华为云 ModelArts 云平台进行模型的训练，得到深度学习的调制类型盲识别模型，然后基于国产开发板 OrangePi Aipro 对模型进行端侧部署，实现了智能通信中调制类型盲识别系统从搭建、训练、推理到边缘部署的全套流程。

本实验的数据处理、算法开发、模型训练、模型部署等环节都可以在 ModelArts 上完成，它支持 Tensorflow、PyTorch、MindSpore 等主流开源的 AI 开发框架，也支持开发者使用自研的算法框架，匹配开发者的使用习惯。以下是 ModelArts 的开发和部署流程：

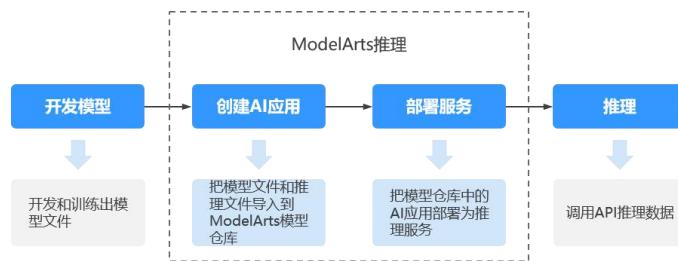


图 13 ModelArts 开发和部署流程图

实验参考论文“Over-the-Air Deep Learning Based Radio Signal Classification”选用卷积神经网络(CNN)开发调制类型盲识别模型，实验所用 CNN 网络包含四个二维卷积层、三个全连接层和一个归一化指数函数(Softmax)，以下是模型结构图：

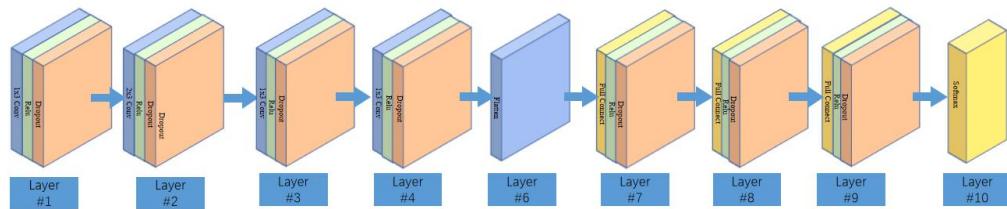


图 14 模型结构图

针对不具备云端模型部署条件的场景，可将模型迁移至端侧设备进行端侧模型推理。在进行模型部署之前，需要使用 ATC 工具对调制类型盲识别模型转换，以适配在 OrangePi Aipro 开发板上的模型部署和推理。ATC（Ascend Tensor Compiler）是异构计算架构 CANN 体系下的模型转换工具，它可将 MindSpore 框架下的的网络模型转换为昇腾 AI 处理器支持的离线模型。以下是 ATC 工具的架构图：

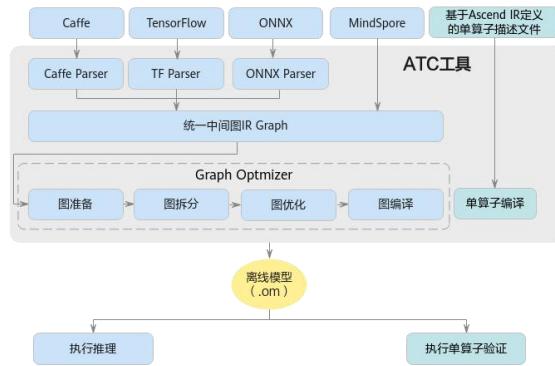


图 15 ATC 架构图

执行 ATC 模型转换指令获得名为 CNN.om 的模型，以供推理使用：

```
(base) HwHiAiUser@orangepi1apro:~/Desktop$ atc --framework=1 --model=CNN.air --output=CNN --input_shape="x:1,1,2,1280" --soc_version=Ascend310B4
ATC start working now, please wait for a moment.
...
ATC run success, welcome to the next use.
```

图 16 ATC 模型转换图

在将模型统一转化为.om 的模型后，还需要掌握 AscendCL 的编程方法以使用 OrangePi AIpro 开发板的 AI 推理功能，在开发板上实现模型的部署。AscendCL 是昇腾计算开放编程框架，下图为 AscendCL 的应用开发流程：

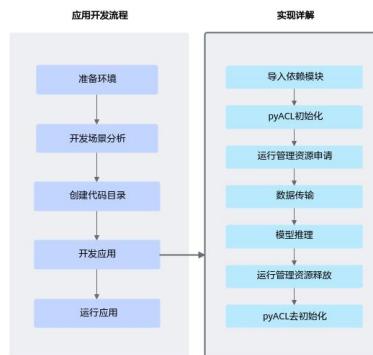


图 17 AscendCL 的应用开发流程图

基于以上的流程完成调制类型盲识别模型端侧部署后，就可以对其模型推理功能进行性能测试。首先对开发板进行初始化，确保硬件资源能够正确运行。然后对模型进行加载，OM 模型文件通过 AclLiteModel 类加载，并对其进行内存分配。OM 模型加载完成后，开发板加载数据执行推理。推理完成后，释放设备资源，Ascend 设备的计算流、上下文及相关资源被正确地清理和重置，以确保系统的稳定性。

以下是性能测试的结果：

```
in inference, use time:0.004334449768066406
Input data shape for instance 998: (1, 1, 2, 1280)
in inference, use time:0.004299640655517578
Input data shape for instance 999: (1, 1, 2, 1280)
in inference, use time:0.004369020462036133
post process
Accuracy: 0.91
process end
acl resource release all resource
AclLiteModel release source success
acl resource release stream
acl resource release context
Reset acl device 0
Release acl resource success
```

图 18 性能测试的结果图

上图显示调制类型盲识别模型识别准确率为百分之九十以上，可以对四种调制类型的数据进行准确的识别，说明模型识别准确率较高，端侧部署成功。

(3) 基于鸿蒙开发板的开源项目经历

Hi3861 是华为推出的一款低功耗、高集成度的 WiFi SoC (System on Chip) 芯片，主要用于物联网 (IoT) 设备，并且对鸿蒙系统高度适配。这款芯片专为智能家居、智能穿戴设备、以及其他小型智能设备设计，支持 WiFi 连接和多种物联网应用。以下是基于 Hi3861 的开发板构成：

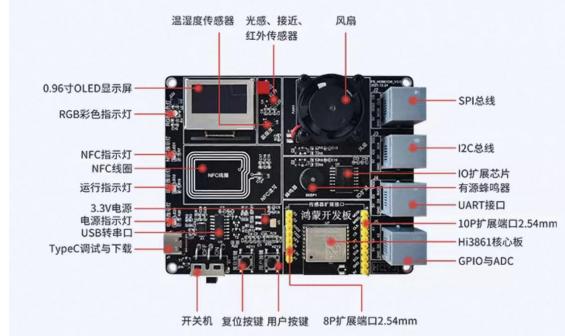


图 19 Hi3861 的开发板构成图

基于搭载 Hi3861 芯片的开发板，设计并开发了智能家居控制系统，通过使用智能灯泡、插座、传感器等设备，实现了远程和自动化控制。以下是在终端通过微信小程序进行远程控制的界面：

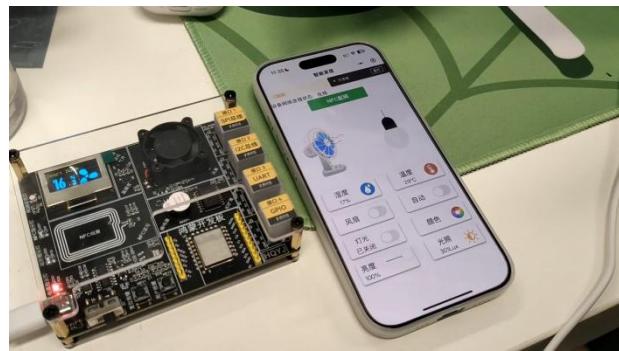


图 20 智能家居控制系统图

以上的功能主要包括通过温湿度传感器采集的数据，自动的调整风扇的工作状态，以及通过远程控制调节智能灯泡的颜色、亮度等。

Hi3861 的设计使得开发板具有低功耗特性，延长了设备的使用时间。其芯片集成了 WiFi 模块、处理器、内存和其他必要的硬件，减少了对外部组件的需求。开发板提供 2.4GHz 频段的 WiFi 连接，支持多种 WiFi 标准，确保设备可以连接到大多数无线网络。它还具有丰富的接口包括 GPIO、I2C、SPI、UART 等，方便与其他传感器或设备进行通信。

基于以上的特性，同样可以使用基于 Hi3861 芯片的开发板制作智能网联车。这里采用成熟的开源开发方案，搭建具有自动避障以及远程控制功能的智能车，并且开发了两种控制软件以适配鸿蒙终端设备或者其他开发系统的终端设备。

首先，针对鸿蒙终端设备控制智能车，使用华为推出的 DevEco Studio 开发工具，通过 ArkTS 语言编写上位机用户控制界面，控制界面上包含了小车的各种功能实现，界面实物图如下：

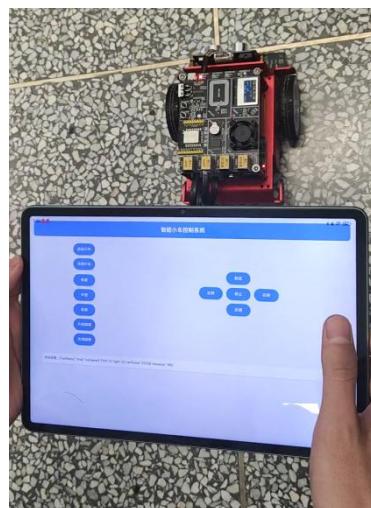


图 21 鸿蒙终端上的智能小车控制界面图

另外，针对其他操作系统的终端设备控制智能车，通过开发微信小程序作为上位机用户控制界面，控制界面同样包含了小车的各种功能实现，界面实物图如下：

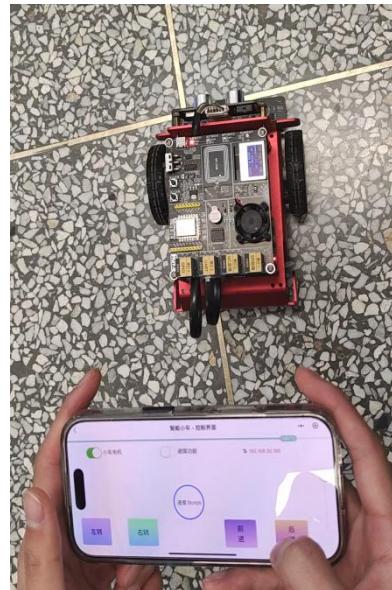


图 22 微信小程序上的智能小车控制界面图

三、项目规划

1. 项目第一阶段（2025 年 7 月 1 日 - 2025 年 7 月 15 日）：

目前已经对智能小车的搭建有了初步的规划，接下来搜集交通领域的目标检测相关数据集、文献、源码，深入学习目标识别以及车道检测算法相关知识，学习多种编程语言实现 Orange Pi AIpro 开发板的模型部署和鸿蒙系统下的软件开发，以方便后续的工作。

2. 项目第二阶段（2025 年 7 月 16 日 - 2025 年 8 月 16 日）：

第一、二周的主要工作是对目标检测功能和车道检测和偏移量计算功能进行复现，并对智能小车的运动和控制逻辑进行实现。

第三、四周的主要工作是尝试将算法、模型等部署至 Orange Pi AIpro 开发板，测试智能小车的目标检测功能以及车道线检测功能，并根据获取到的车道检测信息，对车道偏移计算算法进行设计和优化。

3. 项目第三阶段（2025 年 8 月 17 日 - 2025 年 9 月 10 日）：

该阶段主要工作是对小车的组件进行组装以及对运动和控制功能进行测试，实现控制者通过 wifi 远程控制智能小车运动，并解决视频数据的推流问题，使得智能小车采集的实时视频可以传输到远端。

4. 项目第四阶段（2025 年 9 月 11 日 - 2025 年 9 月 18 日）：

在具有 wifi 通信功能的鸿蒙终端上开发应用程序，实现在鸿蒙终端上对智能小车的远程遥控。

5. 项目第四阶段（2025 年 9 月 19 日 - 2025 年 9 月 28 日）：

对智能小车进行联调，对比开发需求测试智能小车的各项功能，并完成结项报告。

四、期望

希望借助开源之夏这个机会，作为开发者参与到开源项目中，积累开发经验，提升实战能力，为日后参与更多开源项目提供一个经验借鉴。此外我对深度学习以及各种软硬件开发非常感兴趣，开发经验比较充足，希望能在本次开源项目的活动中做出力所能及的贡献。