****

**学生研究计划(SRP)项目验收**

**项目成员个人结题报告**

|  |  |
| --- | --- |
| **参加项目名称：** | **基于深度学习的移动机器人自主导航系统** |
| **参加项目编号：** | **X202210561839** |
| **参加起止时间：** | **2022年 4 月至2023年 3 月** |
| **指导教师姓名：** | **占宏** |
| **指导教师所在学院：** | **自动化科学与工程学院** |
| **学生姓名：** | **韩宇翔、张晓逊、胡宇凡、尹博** |
| **学生学号：** | **201930360367、201930364334**  **202030460414、202130300771** |
| **学生手机号：** | **13923131342** |
| **学生所在学院：** | **自动化科学与工程学院** |
| **学生所学专业：** | **智能科学与技术** |
| **填表日期** | **2023年 3 月** |

|  |  |
| --- | --- |
| **教务处** | **制** |
| **2022年3月** |

**撰 写 说 明**

**1．本报告的文件名请修改为：学号（注意文件名不要出现中文）**

**2．项目所有参与学生分工协作，合力完成一份结题报告，学生在系统提交个人报告时可通用，也可在结题报告基础上适当修改后再提交个人结题报告。**

**3．本“项目成员个人总结报告”建议提纲：（一）绪论**（包括研究背景、国内外研究现状等）、**（二）相关理论知识、（三）****研究方案、（四）实验结果**（包括实验方法、环境、结果对比分析等）、**（五）研究工作总结与展望**（项目主要工作总结及本课题下一步研究工作设想）、**（六）个人总结**（项目组成员分工，本人主要工作和成效，个人参与科研工作的感想等）、**（七）参考文献。**

**注意：**项目成员分工及本人主要工作和成效要细化。相关成果如论文首页，专利/软件著作权，竞赛获奖证书等扫描附在本文档后。

**4．原则上要求本研究报告不少于2500字，相关成果可以成员间重复使用。**

**目录**

[1. 绪论 1](#_Toc30936)

[1.1. 研究背景 1](#_Toc9876)

[1.2. 国内外研究现状 1](#_Toc3111)

[2. 相关理论知识 2](#_Toc26761)

[3. 研究方案 3](#_Toc24518)

[3.1. 机器人总体框架 3](#_Toc31185)

[3.2. 硬件系统 4](#_Toc32152)

[3.3. 软件系统 4](#_Toc18683)

[3.3.1. Android App 4](#_Toc15709)

[3.3.2. 网络架构 6](#_Toc3207)

[3.3.3. 损失函数 6](#_Toc19021)

[3.3.4. 验证度量 7](#_Toc16028)

[3.4. 训练环境 7](#_Toc2684)

[4. 实验结果 8](#_Toc22371)

[4.1. 网络训练结果 8](#_Toc7754)

[4.2. 真实场景测试 11](#_Toc17196)

[5. 研究工作总结和展望 13](#_Toc8306)

[5.1. 工作总结 13](#_Toc15050)

[5.2. 未来展望 14](#_Toc21366)

[6. 个人总结 14](#_Toc6091)

[6.1. 项目组成员分工 14](#_Toc3954)

[6.2. 个人工作情况与感想 15](#_Toc19320)

[(1) 韩宇翔 15](#_Toc7081)

[(2) 张晓逊 16](#_Toc5820)

[(3) 胡宇凡 17](#_Toc14657)

[(4) 尹博 17](#_Toc21095)

[参考文献 19](#_Toc4358)

1. **绪论**
   1. **研究背景**

传统的移动机器人在一个相对稳定的场景下工作，其工作内容往往是单调重复的，而如今移动机器人应用越来越广泛，与人类的交互越来越多，其应用场景也越来越复杂多变。比如在公共场合中负责导航的移动机器人，它不仅需要向人类提供导航服务，带领人类到达目标点，还需要躲避环境中的行人或是正在移动的物体。要使移动机器人能够有效地与人类交互，并且快速适应复杂的工作环境，其必须具备准确的识别人类意图的能力以及良好的自主导航能力。然而，目前市面上设计的满足上述要求的大多数移动机器人都装备昂贵的传感器，如激光雷达和深度相机，高部署成本是限制其大规模应用的原因之一。本项目着眼于解决机器人开发与部署成本高，算法复杂，难以大规模应用的问题，利用深度学习的方法代替价格高昂的高精度传感器。项目主要是将深度学习算法与自主导航相结合，设计并开发基于深度学习的移动机器人自主导航系统。

* 1. **国内外研究现状**

目前已经有不少在降低机器人成本的方向上努力的工作。例如在四足机器人的研究中，Kau等人[1]和Grimminger等人[2]提出了低成本的四足机器人设计方案，依靠低成本的驱动器和零件或3D打印技术，将上万美元的成本降为3千美元和4千美元，但仍保持着和高成本四足机器人相似的功能。在机械臂方面，Yang等人[3]、Gupta等人[4]和Gealy等人[5]分别提出了成本为2千美元、3千美元和5千美元的机器人。另外，AWS DeepRacer[6]、DJI Robomaster S1[7]、Nvidia JetBot[8]和DukieBot[9]，为业余爱好者和研究人员推出了一系列价格在250至500美元之间的移动机器人。但现实是为了达到这个价位，这些机器人的设计就必须在物理结构、感知、通信和计算方面做出妥协。像Nvidia JetBot和DukieBot目的是最大程度保证传感器和机载计算机的性能，但伴随而来的缺点就是又小又慢。

近年来，基于深度学习的自主导航方法逐渐兴起。这种方法使移动机器人可以根据传感器的感知输入直接计算出导航控制命令，也被称为端到端控制。比较有代表性的例子有1989年的ALVINN自动驾驶汽车，2004年DARPA提出的DAVE，Bojarski等人在2016年提出的用于自动驾驶的端到端架构网Pilotnet[10]。

假如按照传统的方式搭建机器人，在机器人独立配置激光雷达，深度相机，机载计算机这样的设备，基本难以做到性能高的同时而不去降低某些部分的功能，因此需要在结构上做一些创新。最近，研究者将端到端的自主导航方法与智能手机相结合设计了智能小车Openbot[11]。M. Müller等人提出的Openbot开创性地在移动机器人上使用手机，使用深度学习算法进行移动机器人的自主导航，降低了机器人对高精度传感器的需求，这使得在保持低成本的的同时也能够保持移动机器人优越的性能。

本项目研究的主要内容就是基于Openbot进行开发，搭建简单低成本的的移动机器人平台，使用智能手机作为机器人的感知和规划控制中心，目标是实现一套只要有智能手机，每个人都能即插即用的移动机器人自主导航系统。

1. **相关理论知识**

深度神经网络是本项目的核心内容，由于深度神经网络的研究内容十分丰富，这里仅阐述本项目中使用到的内容。

卷积神经网路(CNN)是在图像特征提取和分类领域中使用非常广泛的网络架构。目前的卷积神经网络一般是由卷积层、汇聚层和全连接层交叉堆叠而成的前馈神经网络，使用反向传播算法进行训练。

卷积神经网络有三个结构上的特性：局部连接，权重共享以及汇聚。这些特性使得卷积神经网络具有一定程度上的平移、缩放和旋转不变性。局部连接指的是每一层网络中的每个神经元只和下一层的某个局部窗口内的神经元相连，构成一个局部的连接网络。权重共享指的是作为参数滤波器的卷积核对于同一层中的所有神经元都是一样的。卷积层的作用是提取一个局部区域的特征，不同的卷积核相当于不同的特征提取器，每一层之间的传递关系见公式(1)和(2)。

 ( )

 ( )

其中Wp是当前层的三维卷积核，Xp-1是上一层的输出特征映射，bp是当前层的偏置，Zp是当前层的卷积后的结果，之后经过非线性激活函数后得到当前层的特征输出Xp。

汇聚主要由汇聚层(Pooling Layer)操作，一般可进行最大池化下采样和平均池化下采样的操作。典型的汇聚层是将每个特征映射划分为2×2大小的不重叠区域，然后使用最大汇聚的方式进行下采样。汇聚层可以看做一个特殊的卷积层，卷积核大小为m×m，步长为s×s，卷积核为max函数或者mean函数。过大的采样区域会急剧减少神经元的数量，会造成过多的信息损失。

在经过多层的卷积核汇聚操作之后，图像特征会被缩放到一个比较小的尺寸，此时就会将特征的二维矩阵拉平为一维向量，然后使用全连接层对一维向量进行拟合得到最终的网络输出O，O可以是一个一维向量，也可以是由one-hot向量组成的矩阵，其维度由数据类别来决定。神经网络的训练过程就是将每张图像从网络的输入端传入，利用网络的输出O和每个图像的真实类别构建损失函数进行优化，整个深度神经网络在训练过程会使用反向传播算法来不断优化每一层的权重。

1. **研究方案**

本项目在开源项目Openbot框架的基础上进行进一步开发，避免重复造轮子的情况。底层利用Arduino微控制器、电机以及传感器等搭建移动机器人硬件平台；上层利用智能手机作为机器人的大脑，控制机器人实现各种高级任务。将手机端获取的RGB图像，电机的速度信息等传感器信息经过预训练的神经网络，以网络的输出信号作为机器人的导航控制信号，从而避免复杂的特征工程和规划策略，提高系统导航的自主性。

* 1. **机器人总体框架**

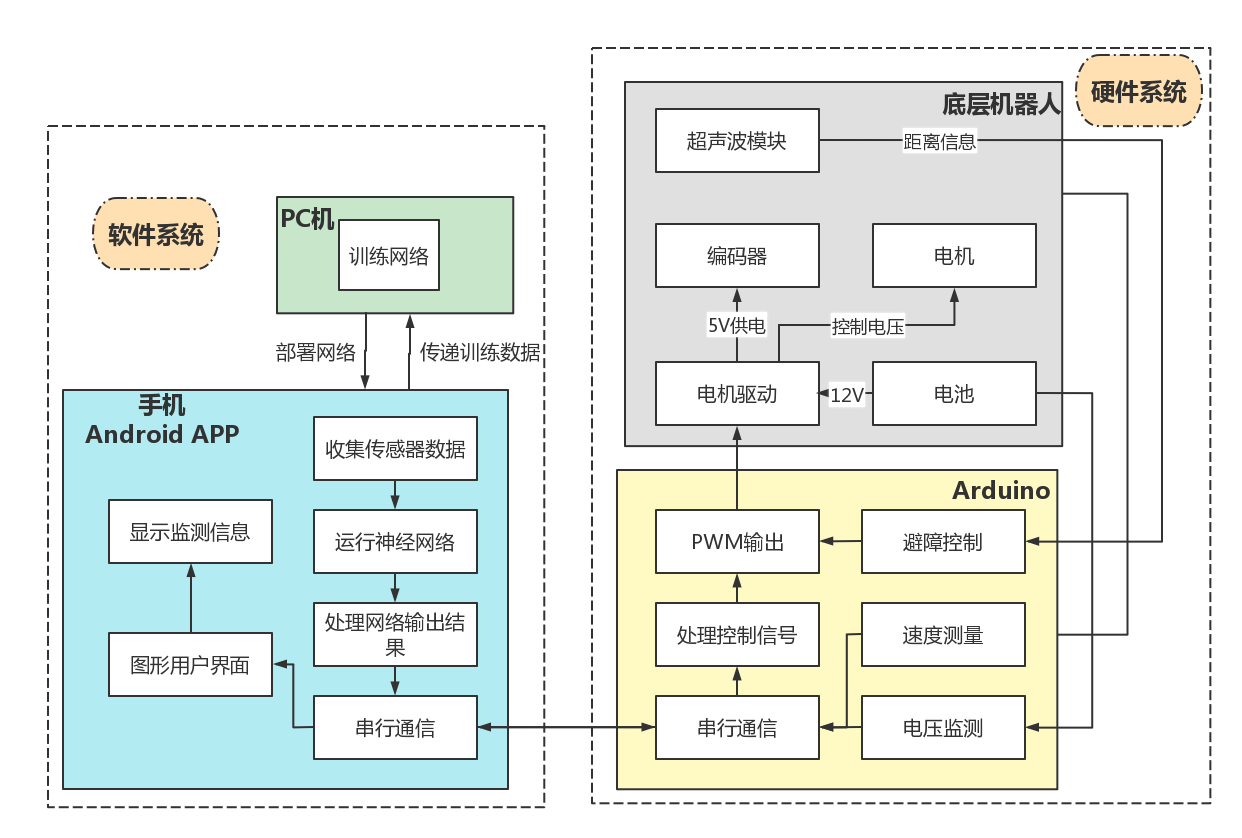
****

图 1 本项目设计的移动机器人的系统框架图

图1展现的是本项目设计的机器人系统架构。整个系统可分为软件系统和硬件系统。总体而言，通过手机获取训练数据，将训练数据传到到PC机上训练一个工作良好的神经网络模型，将该模型部署到安卓手机上后能通过利用实时获取的图片信息计算得到一个合理的输出。Arduino控制器利用该输出信号来驱动机器人运动而实现人物跟随和自主导航的功能；同时操作者能通过语音交互功能控制机器人的工作模式和获取机器人的工作状态。机器人的工作状态可以在Android手机APP上进行显示。

* 1. **硬件系统**

表 1 移动机器人材料清单

|  |  |
| --- | --- |
| 材料名称 | 数量 |
| 80mm橡胶麦克纳姆轮 | 4 |
| 520编码器直流电机 | 4 |
| 亚克力底板 | 2 |
| 18560锂电池(2000mAh) | 3 |
| BTN7971电机驱动 | 2 |
| Arduino mega2560开发板 | 1 |
| 手机支架 | 1 |

我们搭建的移动机器人所用的材料清单见表1。由于智能手机集成了很多功能，组成底层的移动机器人的材料和结构并不复杂，只由车模、带编码器的电机、电机驱动、车轮和Arduino组成一个移动机器人的最小系统，后面额外添加了一个超声波测距模块用以紧急制动。

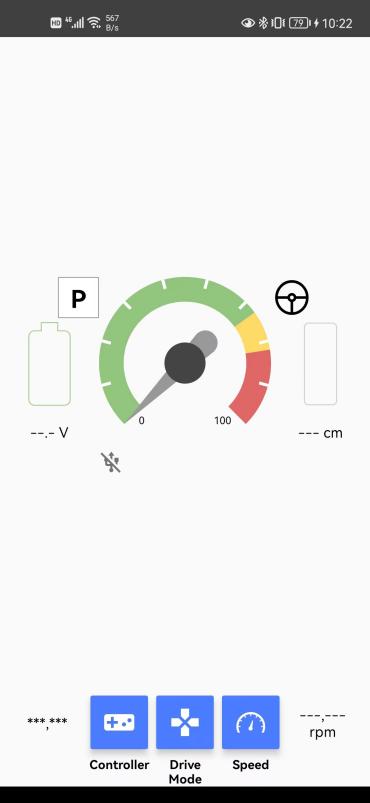
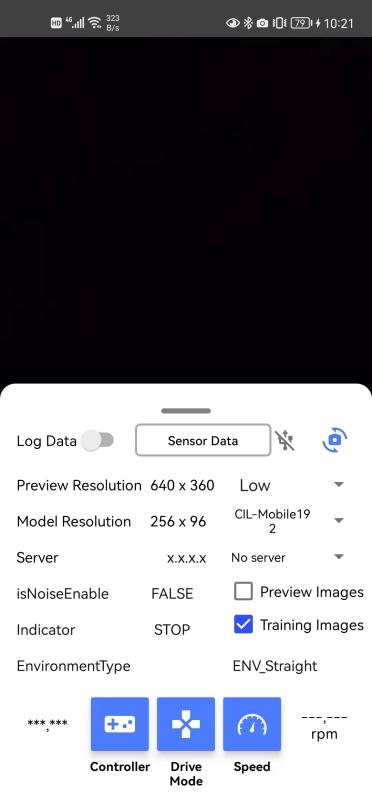
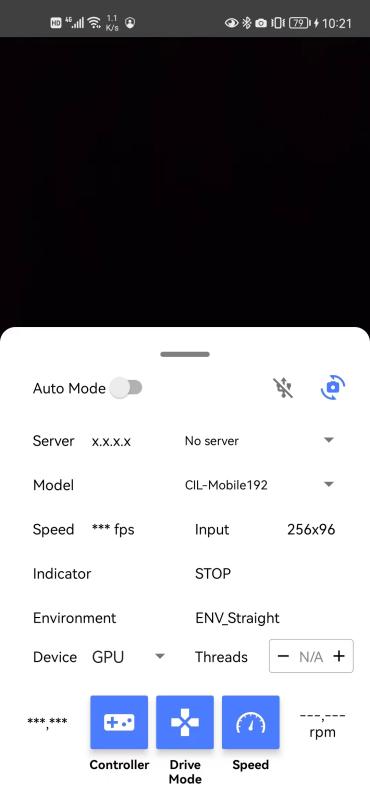
Arduino的工作也比较简单，首先它需要保持和智能手机通讯，将手机传过来的控制信息，转化成电机控制信号输出；其次需要不断监测机器人的运动速度、超声波的距离信息和电池电压，确保移动机器人的行驶安全。

* 1. **软件系统**

本项目的软件系统是在Openbot原有的基础上进一步改进，使系统更加符合项目的任务需求。

* + 1. **Android App**

Android App提供了一个统一的界面，可以从任何Android智能手机获取传感器数据。App构建了一个数据记录器，以便用机器人收集数据集。目前，App可以记录手机中原有的传感器的数据和车身传感器的数据（轮式里程计、和电池电压），这些数据通过串口通信传输。本项目需要采集的数据有：手机相机采集的RGB图像数据和操作者对机器人的操控数据，其中操控数据包括转向信号和电机输出值。

(a) (b) (c)

图 2 Android App界面

图二是我们使用的Android App的主要界面。图2(a)主要用来测试与手机的通讯是否正常，若通讯正常，机器人收到的传感器信息会在这里正常显示。图2 (b)是我们收集数据使用的界面，上面部分显示摄像头的实时画面，下面部分是相关参数的设置，可以调节图像分辨率，设置为哪一个神经网络收集数据等。图2(c)是机器人进行自主导航任务时运行的界面，上面部分同样显示摄像头的实时画面，下面部分也是相关参数的设置和部分标志位的显示，我们可以选择使用哪一个神经网络，是否使用GPU等参数，这里同样会接受并可视化机器人传来的传感器信息。

本项目会将训练好的神经网络部署到智能手机上，利用智能手机的计算能力来处理传感器数据输入并实时计算机器人的动作。特别的，我们使用Tensorflow Lite来表示我们的模型，它能够无缝集成到Android App中，部署过程很简单。在PC机中使用Tensorflow训练网络模型后，将其转换为Tensorflow Lite模型，可直接部署在智能手机上。神经网络的预测都只依赖于摄像机的反馈，无需轮式里程计或超声波传感器读数即可生成原始车辆控制信息。

* + 1. **网络架构**

这里所使用的神经网络模型与一般的图像分类模型的架构并不一样，我们需要的训练数据除了RGB图片，还有数据采集过程中操作者的控制信号，这个控制信号是告诉机器人遇到分岔路口时候要往哪个方向走。因此，我们使用的模型是两个网络的组合体。

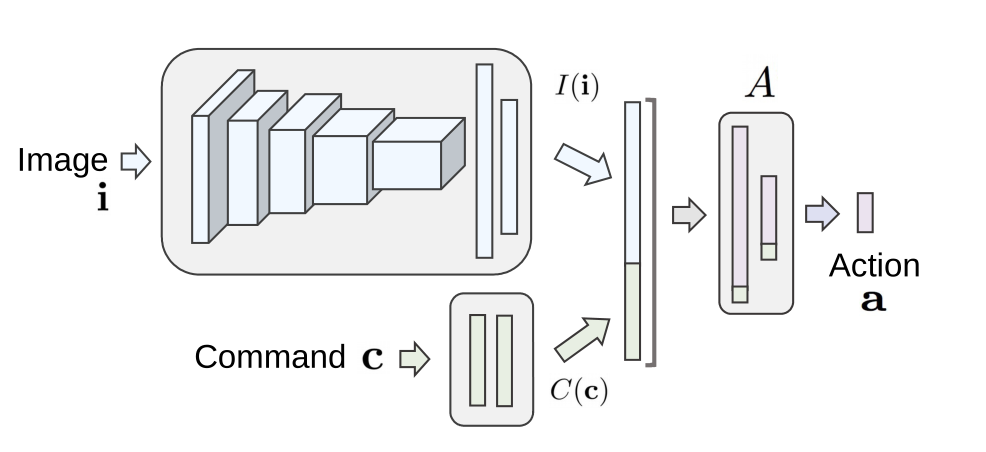


图 3 自主导航中的神经网络架构

这是一个类似于条件模仿学习的命令输入变体的神经网络[12]。该网络比现有网络小大约一个数量级，甚至可以在中端智能手机上实时运行。

网络结构如图3所示。它将图像i和命令c作为输入，并通过图像模块I(i)和命令模块c(c)处理它们。图像模块由五个卷积层组成，分别具有32、64、96、128和256个卷积核，每个卷积核的步长为2，第一层的核大小为5，其余层为3。我们在每个卷积层之后应用relu激活函数、Batch-Normalization和20%的Dropout。输出由128和64个单元的两个全连接层拉平和处理。命令模块被实现为具有16个隐藏神经元和16个输出神经元的MLP。图像模块和命令模块的输出被拼接成一个一维向量并输送到控制模块A，控制模块A也是MLP，包含两个分别具有64个神经元和16个神经元的隐藏层，最后被映射到动作向量a。这个动作向量a就是要传递给Arduino控制机器人运动的。

* + 1. **损失函数**

由于我们的任务与普通的图片分类并不一样，因此需要使用新的损失函数和验证度量来训练该网络。

在自动导航数据集上训练端到端驾驶策略时，一个常见的挑战是巨大的标签失衡。大多数情况下，车辆都是直线行驶，导致许多图像带有相同的标签。

一种常见的方法是在训练过程中对数据集重新采样或仔细制作单个批次[12]。然而，这通常需要固定的数据集或计算开销。如果数据集是动态变化的或以连续流的形式到达，则这些方法是不可行的。

我们可以通过加权损失来解决这种不平衡。思想很简单：转向角越大，操纵越关键。因此，我们使用加权项与转向角成比例的损失，以及整个动作作用向量上的MSE Loss，以确保网络也能学习油门的输出值。

 ( )

 ( )

 ( )

其中，at是目标动作，ap是预测动作，st是目标转向角。ω是权重，以调整转向角损失的重要程度，具有偏置b，以控制具有零转向角的样本的权重。由于我们的车辆使用差速转向，动作向量由二维控制信号a=(al, ar)组成，对应于左右车轮的油门输出。转向角s就是左油门和右油门的差值。

* + 1. **验证度量**

训练自动驾驶策略并根据训练或验证损失对其进行评估的一个主要挑战是与驾驶策略的最终性能缺乏相关性[13]。不同的动作序列可以导致相同的状态。验证损失度量目标和预测之间的相似性太严格了。因此，这里定义了两个较不严格的验证指标，并减少了离线和在线评估之间的差距。第一个度量是度量转向角是否在给定阈值内。第二度量甚至更宽松，并且仅考虑目标的转向方向和预测方向是否对齐。我们根据验证集上这两个度量的平均值来选择最佳检查点。

* 1. **训练环境**

收集网络训练数据的环境是一个回字形走廊，如图4所示。整个走廊可以分为四个路段，各路段中间均有地毯，可以天然作为图像的一个特征，每一段的具体情况可见图5。



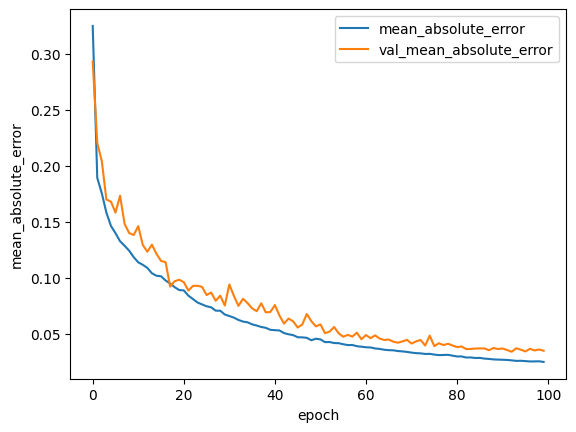
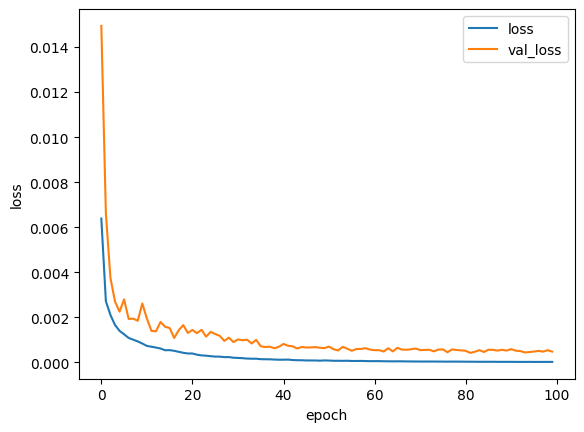
图 4 总体训练环境



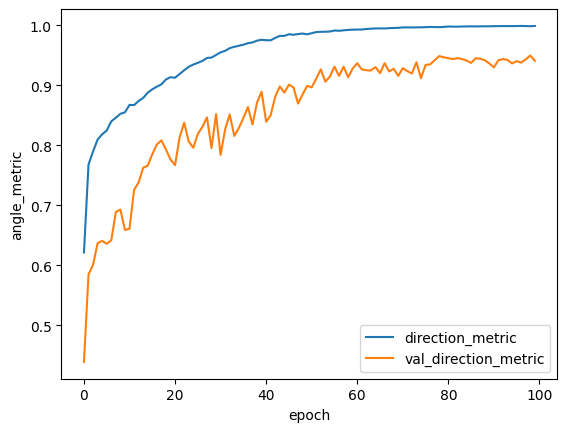
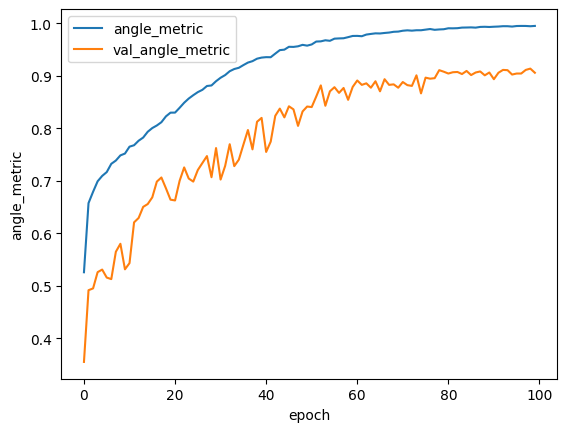
图 5 各路段具体情况

路段二中部存在一段亮度较低的区域，我们需要在后面的实际场景测试中特别关注这一段的运行效果。为了更好地测试小车的自主导航效果，我们在路段二的最后部分布置了几个障碍，在实际场景测试中评估小车是否能够复现训练时的避障动作。

1. **实验结果**
   1. **网络训练结果**

****

1. (b)

****

(c) (d)

图 6 训练结果可视化

我们的数据集划分比例大概为训练集:验证集:测试集=8:1:1。训练过程中主要的超参数设置为：batchsize是128，学习率lr为0.001，训练轮数epoch为100。同时也对图像进行了一定的颜色域上的数据增强。训练结果见图6。图6(a)是损失函数的图像，训练损失和验证损失的值不断下降且没有出现明显的震荡，说明网络优化效果较好，并没有出现过拟合的情况。图6(b)是网络输出动作向量的均方误差的图像，这个描述的是网络学习油门控制的效果，MSE的不断下降也说明网络较好的学习了操控员的油门控制。图6(c)是角度度量，这个值越大说明网络学到了操控员的转弯角度控制，机器人能够以一个更加合适的角速度来执行转弯的动作。图6(d)是方向度量，这个值越大说明发生转向错误的情况越低。从训练曲线上看，网络的训练结果是朝着我们期待的方向进行的。

图7和图8、图9是对测试集的测试结果。在图7直走情况下，网络的预测也是直走，而且网络输出的油门控制效果也和数据集收集过程中操作员的控制效果相当；图8的转弯情况下，网络也能输出相应的控制转弯的油门控制；图9临近转弯的情况下，即使我们先提前给出了转向信号，网络的输出也不会马上输出转弯的油门控制，而是直到快要到达转弯处才会逐渐输出转弯的油门控制。以上的三种情况有效地说明我们的网络训练效果是比较好的，符合我们的预期效果。

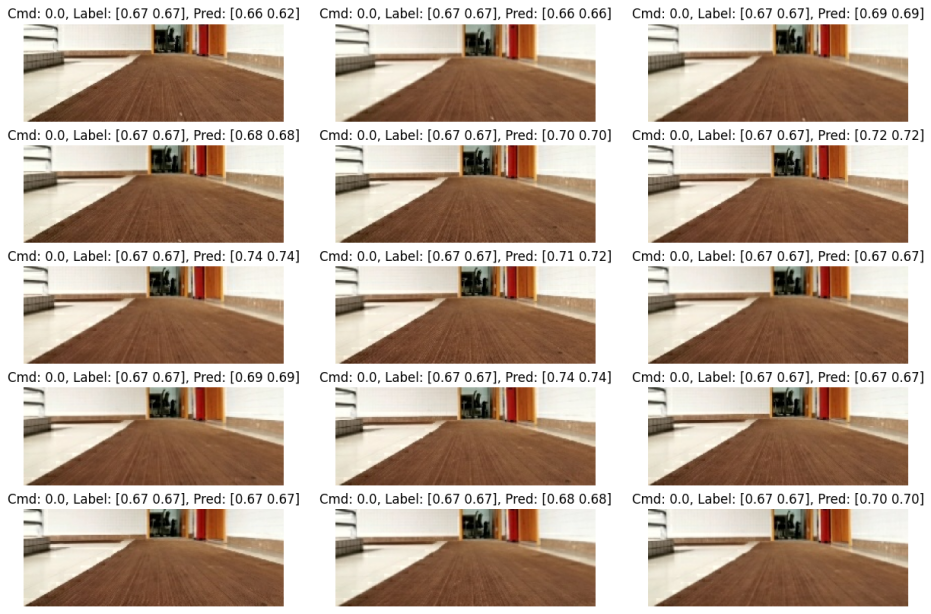
****

图 7 网络预测-直走情况

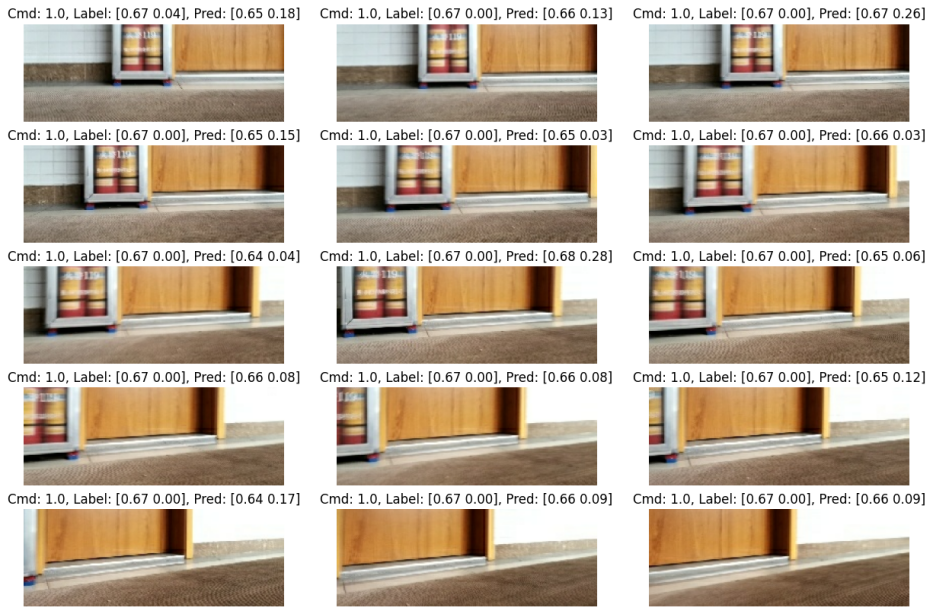
****

图 8 网络预测-转弯情况

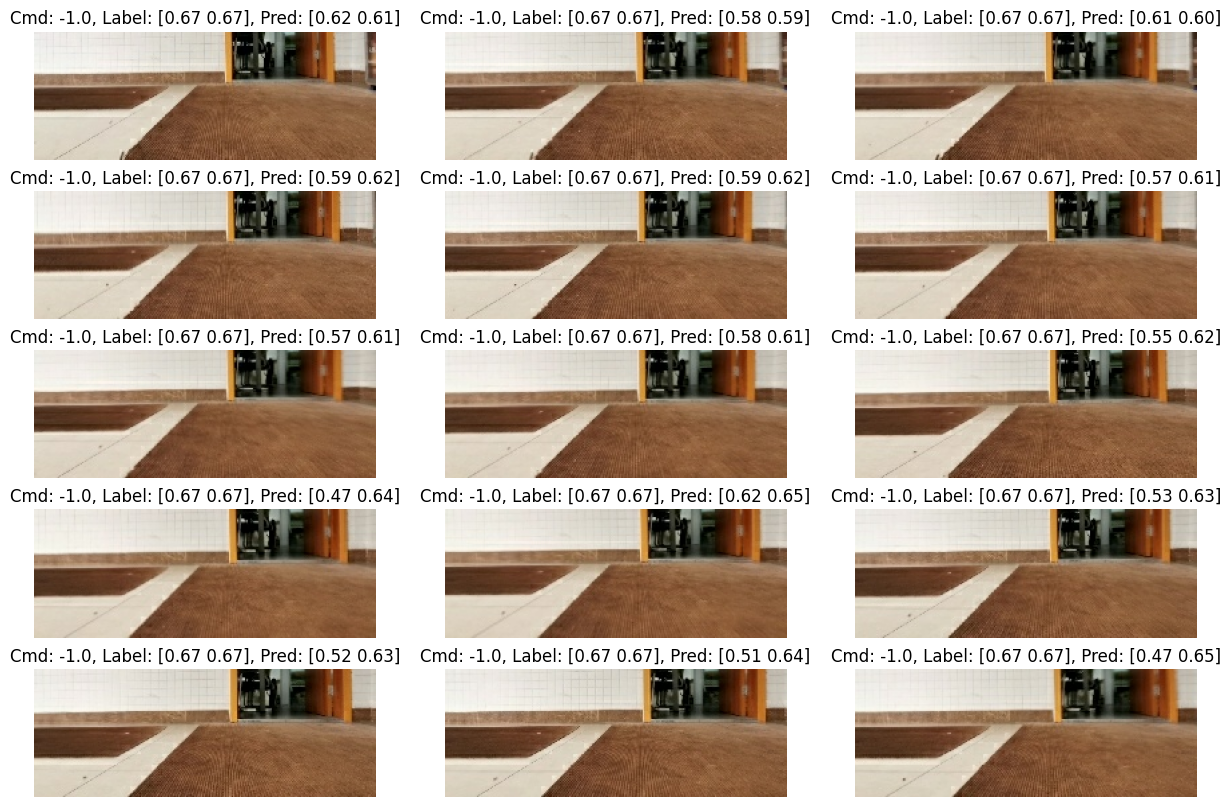
****

图 9 网络预测-临近转弯情况

* 1. **真实场景测试**

将上述训练好的网络部署到智能手机上后，我们回到原训练环境进行实际场景测试。图10、11、12展示的是机器人在实际场景测试中的运行状态，左半部分是机器人上的手机摄像头视角，右半部分是外部视角。图10所示的直行效果较好，小车可以大致保持在地毯中心线运行。但在路段二亮度较低的部分，小车的运动出现无法预测的情况。我们推测这部分的图片因为亮度太低，并未被网络归类为直行情况。转弯时，图11中所示的路段四左转转向路段一的效果较好。小车大多数情况下可以按照训练时的转弯策略自主转弯，但在某些路口的特定方向，仍需要人为给予转向控制信号。关于避障效果，我们训练时的驾驶策略是小车尽量远离障碍行驶，但尽量不驶出地毯。从图12中可以看出，小车较好地复现训练时的驾驶动作。

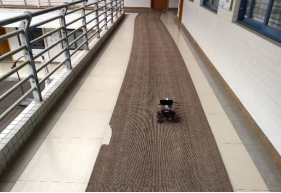
** **

图 10 直行效果手机视角（左）和外部视角（右）

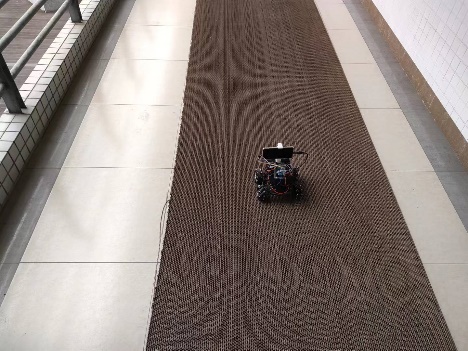


图 11 转弯效果手机视角（左）和外部视角（右）

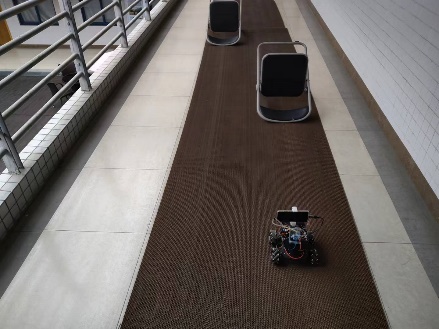
 

图 12 避障效果手机视角（左）和外部视角（右）

1. **研究工作总结和展望**
   1. **工作总结**

在近一年的时间里，小组利用课余时间，对课题“基于深度学习的移动机器人自主导航系统”进行了充分的实践，主要完成了以下工作：

1. 搭建了一台基于Arduino单片机和麦克纳姆轮底盘的移动机器人；
2. 实现了机器人底层的运动控制方案和与智能手机的通信接口；
3. 基于Openbot的App进一步开发，搭建了一个更加契合项目研究内容的App；
4. 实现了Openbot的原生策略网络，并在训练策略中加入了一些当前使用比较广泛的tricks，提高了训练效率和质量。
5. 在真实场景完成项目测试。
   1. **未来展望**

目前这个系统已经基本能够复现Openbot中所提到的自主导航功能。无论是在训练结果上，还是到真实场地中测试，试验结果说明神经网络的确可以实现机器人的自主导航。

当然目前这个功能还并不完善，根据Openbot的网络结构，机器人在遇到分岔路口的时候是需要更上层的规划模块给出具体的前进方向的，也即图2中网络的第二个输入c。如果这个c没有指定的话，有可能出现病态的运动情况，比如左右摇摆、只往前走等。关于这个问题目前有几个可行的方案：在检测到有分岔路口时，简单地随机分配一个转向信号c，亦或是增加一个更上层的规划模块，按照实际需求指定一个转向信号c。后续要完善这个功能才能使得这个自主导航系统更加完备。

在功能的多样性方面，我们考虑可以加入语音控制的相关内容，让我们的机器人变得更加智能，人机交互更加方便。

我们相信，经过不断的完善，我们的系统能够应对广泛的场景自主自主导航需求。

1. **个人总结**
   1. **项目组成员分工**

表 2 项目组成员分工情况

|  |  |
| --- | --- |
| 项目组成员 | 任务分工 |
| 韩宇翔 | 负责数据采集和整理、机器人单片机部分功能的代码编写、实际场景测试 |
| 张晓逊 | 负责机器人单片机部分功能的代码编写、数据处理、神经网络的搭建和训练、实际场景测试 |
| 胡宇凡 | 负责移动机器人材料选型、搭建和维护 |
| 尹博 | 负责Android App设计与开发 |

* 1. **个人工作情况与感想**

1. **韩宇翔**

本次项目中，我负责的是单片机部分功能的代码编写，数据的收集和整理以及实际场景测试。

首先，在单片机代码编写部分，我完成了小车速度设定和电机速度控制算法。小车速度通过手柄设定，电机速度控制算法采用增量式离散pid控制算法。代码编写和一般嵌入式编程套路相同，明白相关逻辑后较为容易实现。

其次，数据收集部分，主要是通过手柄人为控制小车运动，收集运动过程中的摄像头数据、小车速度数据等。实际场景测试时，将训练好的神经网络模型重新部署回手机，由网络进行小车运动控制而不人为干预，之后进行系统性能评估。

由于能力有限，在之前的时间里一直没有机会参加合适的科研项目。因此，深知此次参加科研的机会来之不易，我也付出很多精力完成项目中自己负责的部分，当然也收获颇丰。首先，我认识到在科研工作（尤其是理工科相关）中，实践能力十分重要。比如设计电机速度控制算法时，我曾在课堂上学习过许多控制算法，但进行代码实现时，需要考虑实际情况，进行速度限幅、处理编码器数据等。其次，我认识到科研项目分工明确并不代表每个人的工作是完全割裂的。在数据收集时，我需要了解收集什么样的数据、怎样收集数据以及数据如何打包整理。所以，我需要了解小车的机械结构、神经网络的框架结构以及open bot APP的设计,这些都是队友们工作的核心部分。在推进项目的过程中，我们一直保持沟通、交流，我也向队友请教了许多问题，在遇到困难时也获得了队友们以及老师的帮助，体会到了团队力量的强大。

总之，此次项目的经历对于我来说意义价值重大。通过此次项目，让我第一次接触到科研工作，接触了前沿的深度学习算法，让我学习了许多新技能，丰富了实践经历。同时，我十分感谢组内各位队友的合作与帮助以及师兄和老师对我们的指导。我会继续提升自己的能力，持续学习，在科研的道路上勇往直前。

1. **张晓逊**

本次项目中，我负责的是单片机部分功能的代码编写、神经网络的搭建和训练和实际场景测试。

单片机中代码实现部分，我完成了电机驱动、编码器测速、串口通信、超声波测距。这些工作难度不大，主要需要熟悉Arduino的代码设计逻辑和一些传感器的原理。

神经网络的部分保持了Openbot的主体网络架构，我主要是修改了一些训练的tricks，比如添加了Onecycle学习率调整策略，使网络训练效率和效果有了一定提升。另外我还负责数据处理的工作，将一些无效的数据和异常数据清洗掉。

这一年的项目经历还是比较丰富的，在确定选题之后，虽然说我们有了明确的分工，但是还是会不断地沟通、交流，为的就是一起做好这个项目。一年下来，我们完成了这个项目，也学到了很多东西，收获颇丰。

我最大的感受就是参加科研项目会比平常单纯的上课学到更多的东西，因为做项目一定会遇到困难，遇到我们不懂的事物，我们在克服它们的时候就必然会去学习，去探索，渐渐地我们就会掌握到很多课程以外的知识。就拿本次项目来说，我接触到很多新的东西。项目机器人的核心组件是智能手机，用智能手机开发就离不开Java。但在开始这个项目之前我从未接触过Java，我特地花费了很多时间去认真学习了Java。虽然说可能后面不会在这个方向发展，但掌握多一项技能也是很好的。

此外，团队的力量也能给我们很多的帮助。在整个项目的进行过程中，我们会偶尔遇到困局，但因为有队友、朋友之间的相互鼓励，团队的力量，我们总能保持激情，不断克服困难。因此，当我们陷入了困局，发现不了其中的问题的时候，不妨向身边的朋友寻求帮助。有的时候可能是自己的思维束缚了自己，不同的思维之间的碰撞，有可能会让我们瞬间发现问题所在，甚至找到更多思路，激发出更强的火花。

总的来说，这次项目经历是一次非常棒的体验。我认识到新的老师、师兄、同学，学到了新的事物，也明白了如何去做一项科研任务，相信这一年的经历可以给我未来的学习和生活带来非常大的好处。我非常感谢老师、师兄的指导和组内各位成员在这一年来对我的帮助，我也会带着我学到的东西继续努力，为实现我的理想而继续奋斗。

1. **胡宇凡**

本次项目设计中，我负责的是移动机器人底盘搭建的工作，底盘搭建虽然看似不显眼，但其实也是一个很有意思的工作，如何选购硬件，将硬件进行整体的布局，布局的过程需要综合考虑整理线材，美观性和实用性，由于线材选购的种类，选购的配件接线不在同一位置的原因，最终的成品并不算很美观，但是能确保正常使用，也算是勉强完成了任务，参与这一次科研实践活动让我有着许多崭新的认识，科研工作与平日的课程学习差异很大，平日的课程学习可能几个月才有一次检验的机会，反馈的机会少，也容易飘飘然忘乎所以，而科研工作会随着每一步的进程随时给到反馈，或是成功的喜悦，也或是失败的难过，但这种层层的反馈可以一步一步支持着工作者不断前进，不论是从成功中找到方向，还是从失败中积累经验，一段时间的科研工作坚持下来都会让我觉得有着许多的提升。除此之外，科研工作还更加依赖长期的规划和团队协作等综合能力，而这些能力是平时的学习生活中鲜少有机会锻炼的，因此我为有这么一次科研工作感到无比的感激，也为自己坚持到工作的结束感到莫大的光荣。

1. **尹博**

在这个项目中，我有幸与师兄们一起参与了基于深度学习的移动机器人自主导航系统的研究。这个项目不仅让我学会了许多新的技能，同时也让我了解到了科研项目的重要性和工作环境。

首先，在这个项目中，我学会了使用android studio进行app的开发。这是一项非常有用的技能。通过开发这个app，我也学会了如何设计界面、如何与后端交互、如何处理用户输入等方面的技能。这项技能对我未来的职业发展非常重要，因为随着移动设备的普及，这种技能的需求越来越大。

其次，我在这个项目中还学习了许多关于语音处理方面的知识。移动机器人的自主导航需要与用户进行交互，语音处理是其中一个非常重要的方面。通过这个项目，我学会了如何对语音信号进行采集、如何将语音信号转换为文本、如何进行语音合成等方面的技能。这项技能对我未来的职业发展也非常重要，因为随着语音助手和智能音箱的普及，这种技能的需求也越来越大。

此外，在观看师兄们搭建小车的过程中，我也了解了一定的硬件知识。虽然我并不是一个电子工程师，但是这项技能对我理解机器人系统的结构和原理非常有帮助。

通过这个项目，我还有机会亲身参与科研项目，感受到了科研的氛围。我了解了科研的流程和工作方式，学会了如何进行文献调研、如何设计实验、如何分析数据等方面的技能。我还学会了如何与团队合作、如何协调团队成员的工作、如何与指导教师和其他团队成员沟通交流。这些技能对我未来的职业发展非常重要，因为在许多工作中，团队合作和沟通交流是至关重要的。

尽管我的能力不足，没有为团队做出太多贡献，但我在这个项目中付出了努力，从中学到了很多东西。我对于自己不能为团队做出太多贡献感到抱歉，但我会继续努力提升自己的能力，为团队做出更多的贡献。

总的来说，这个项目是一次非常有价值的经历。通过参与这个项目，我不仅学会了许多新的技能，同时也了解到了科研项目的重要性和工作环境。我深刻认识到，只有不断学习和提升自己的能力，才能在未来的职业生涯中取得更大的成就。我也更加珍惜与团队成员的合作机会，努力为团队的成功做出贡献。

在未来的学习生活中，我会继续发扬学习的精神，不断提升自己的能力，为团队的成功做出更大的贡献。我相信，在这个项目中学到的技能和经验将会对我的未来职业发展产生重要的影响，我将用这些技能和经验为实现自己的梦想而努力奋斗。

**参考文献**

1. N. Kau, A. Schultz, N. Ferrante and P. Slade, "Stanford Doggo: An Open-Source, Quasi-Direct-Drive Quadruped," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 6309-6315, doi: 10.1109/ICRA.2019.8794436.
2. F. Grimminger et al., "An Open Torque-Controlled Modular Robot Architecture for Legged Locomotion Research," in IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 3650-3657, April 2020, doi: 10.1109/LRA.2020.2976639.
3. B. Yang, D. Jayaraman, J. Zhang and S. Levine, "REPLAB: A Reproducible Low-Cost Arm Benchmark for Robotic Learning," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 8691-8697, doi: 10.1109/ICRA.2019.8794390.
4. A. Gupta, A. Murali, D. P. Gandhi, and L. Pinto, “Robot learning in homes: Improving generalization and reducing dataset bias,” in NeurIPS, 2018.
5. D. V. Gealy et al., "Quasi-Direct Drive for Low-Cost Compliant Robotic Manipulation," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 437-443, doi: 10.1109/ICRA.2019.8794236.
6. B. Balaji, S. Mallya, S. Genc, S. Gupta, L. Dirac, V. Khare, G. Roy, T. Sun, Y. Tao, B. Townsend, et al., “Deepracer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning,” arXiv:1911.01562, 2019.
7. DJI Robomaster S1, “[https://www.dji.com/robomaster-s1](https://www.dji.com/cn/robomaster-s1),” accessed:2023-03-10.
8. Nvidia JetBot, “<https://github.com/nvidia-ai-iot/jetbot>,” accessed:2023-03-10.
9. L. Paull et al., "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 1497-1504, doi: 10.1109/ICRA.2017.7989179.
10. Bojarski M , Yeres P , Choromanska A , et al. Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car[J]. arXiv, 2017.
11. M. Müller and V. Koltun, "OpenBot: Turning Smartphones into Robots," 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 2021, pp. 9305-9311, doi: 10.1109/ICRA48506.2021.9561788.
12. F. Codevilla, M. M¨uller, A. L´opez, V. Koltun, and A. Dosovitskiy, “End-to-end driving via conditional imitation learning,” in ICRA, 2018.
13. Codevilla, F., Lopez, A. M., Koltun, V., & Dosovitskiy, A. (2018). On offline evaluation of vision-based driving models. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 236-251).