**第十七届全国大学生智能汽车竞赛**

完全模型组



**技 术 报 告**

学 校 乐山师范学院

队 伍 名 称 乐师逐飞九队

参 赛 队 员 杨登希 罗道军 薛鹏 张鑫 李岩

带 队 教 师 杨继豪 李富钢

提 交 日 期 2022.7.15

第 一 章 引言

1.1 赛事背景

1.2 本参赛队主要工作

1.3 相关技术研究现状

第 二 章 总体方案概述

2.1 系统架构

2.2 创新与优化

第 三 章 功能模块设计与系统成

3.1 车道线采集

3.2 车道线数据处理

3.3 模型采集

3.4 数据处理及模型训练

3.4.1 数据处理

3.4.2 模型训练

3.5 循迹算法

第 四 章 关键算法现

4.1 数据集制作相关算法

4.1.1 数据采集

4.1.2 数据集制作

第 五 章 测试

5.1 测试环境说明

第 六 章 作品总结

6.1 技术创新

6.2 技术路线

6.3 存在的问题及改进方向

1. **引言**

**1.1赛事背景**

2017 年 7 月，国务院印发《新一代人工智能发展规划》，按照规划，我国人工智能的发展将分三步走，并最终实现在 2030 年使中国的人工智能理论、技术与应用总体达到世界领先水平。近年来，随着人工智能特别是深度学习的发展，将深度卷积 神经网络应用在无人驾驶领域，助力无人车实现智能感知、智能决策等关键技术， 其已经成为一大研究热点。

本次智能车完全模型组别的设立，能够场景化地复现基于深度学习的智能汽车 在实际领域中的应用，尤其是在无人的环境中，实现数据采集、数据模型构建、自主识别弯道、无人驾驶验证等多种技术融合的场景。将深度学习技术赋予机器智能行为，为培养创新综合人才提供演练平台，以赛促教，拓宽高校人工智能相关专业的教学内容，提升高校人工智能科技创新能力和人才培养能力。

**1.2 本参赛队主要工作**

本次比赛任务是：在比赛开始后，参赛车辆从起点出发，沿着车道线行驶，同时识别途中的起跑线、施工区、泛行区、坡道、圆环等标志物，并遵照规则实施相应动作，行驶两圈抵达终点后入库停车。为了完成预选赛任务，本参赛队首先充分学习和测试了参赛车辆的硬件和软件平台，其次在软件上实现并优化了端到端的自主循迹，以及基于模型训练的模型标志检测这两个主要功能，并添加了车体实时避障的功能，最后通过模块整合与联调构成完整的智能汽车系统。 本参赛车辆针对第十五届全国大学生智能汽车完全模型组别的相关要求设计，硬件方面采用组委会统一提供I型车模，未对无人车及配件进行改装或替换， 软件方面使用开源深度学习平台飞桨完成模型的训练、推理和部署。

**1.3 相关技术**

本次完全模型组别的比赛任务主要涉及到模型训练及模型检测技术，以及利用二值化、大律法、多线程实现的巡线功能。利用串口将数据送到下位机实现对车的智能控制。

1. **总体方案概述**

**2.1 系统架构**

整车系统包括三个子模块，分别为基于edge board多线程的基础巡线和AI studio的模型训练及检测，下位机控制主板以及驱动板块。

1. 基础巡线，负责为智能汽车行驶提供道路曲率类型信息。其 主要实现路线为：通过摄像头对赛道的黑色边界进行取值送到edge board计算并且将数据通过串口发送给下位机。
2. 模型训练，负责采集基础模型为计算板卡提供信息。主要实现首先 采集模型在赛道各个位置、各个角度上的图片，接着对每张图片进 行相应处理，进行数据清洗，再将一系列这样的数据放在AI studio 平台上进行模型训练得到最终的模型参数；
3. 下位机控制主板以及驱动板块主要控制车子的舵机转向及电机控 制。通过edge board发送的数据进行计算，然后控制舵机以及电机 最终实现小车的正常行驶。

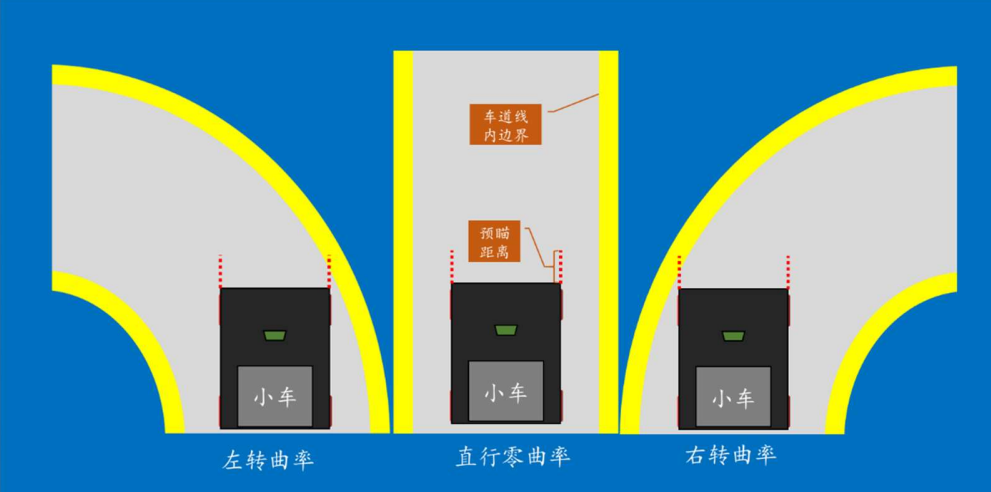
**2.2 创新与优化**

本节主要罗列方案涉及的几个创新点或优化思路。

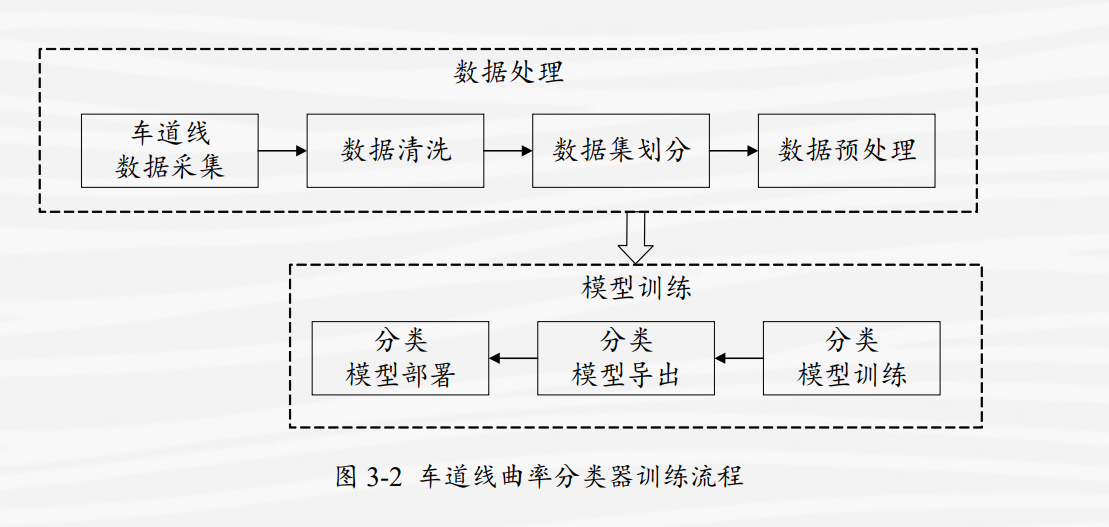
1. 车体循迹方面：将循迹任务看成一个分类任务，在小车行驶时实时 用摄像头采集图片，通过二值化及多线程计算后判断相应参数，用 于后续决策与控制。
2. 模型训练方面：模型训练上舍弃了示例的回归网络，另外设计了一 个三分类网络。
3. 模型采集方面：考虑到小车远程终端无法实时看到摄像头采集的图 片，在 笔记本端编写了相机图片显示脚本，可以在采图前利用笔记 本查看车体相机 图片，确定距离标志物最近和最远的距离，然后再 用小车的采集程序采集。
4. **功能模块设计与系统集成**

**3.1 车道线采集**

由于在车道线检测模型的设计中，将端到端循迹任务看成了一个分类任务，因 此在车道线采集时需要分类别采集不同类别对应的二值化数据。

对于采集数据的类型，如图所示，根据道路曲率划分成如下三类： （1）直行零曲率；（2）左转曲率；（3）右转曲率。

**3.2 车道线数据处理**

为了提升对车道线曲率类别识别的准确率，减少冗余信息对模型训练的影响， 首先对车道线数据进行处理，进而将处理完成的数据送入网络进行训练。此部分整 体流程如图所示。

在经过数据清洗后，将数据按照训练集与测试集为 8：2 的比例对车道线数据继 进行划分，再对原始图像数据进行处理，其具体步骤为：

1. 将采集到的赛道图片，由 RGB 模型转为 HSV 模型；

(2) 根据赛道边界的颜色，在多种光照条件下多次测试，确定提取边界颜色的 gray 上下限；

(3) 使用确定好的 gray 上下限，对转为 gray 模型的图片进行滤波操作，即车道线图片。

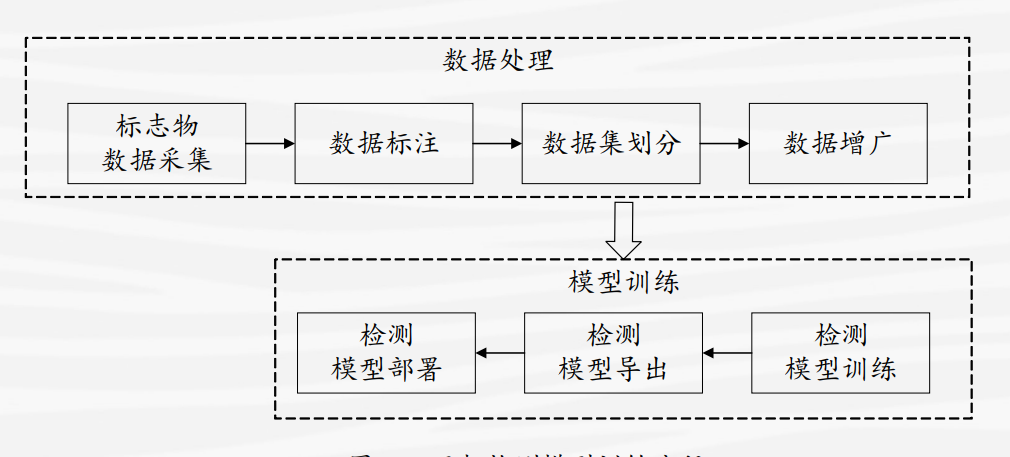
**3.3 模型采集**

本赛道中涉及到的模型主要列举如下：起跑线、施工区、饭行区、坡道、禁行区，如图所示

由于在采集的过程中需要移动智能车至不同的位置，以采集不同的模型和使用不同的姿态采集模型，因此设计的采集模型的程序应该满足如下需求： 

1. 可以方便的暂停与继续采图； 
2. 可以一键删除分类错误的图片序列。

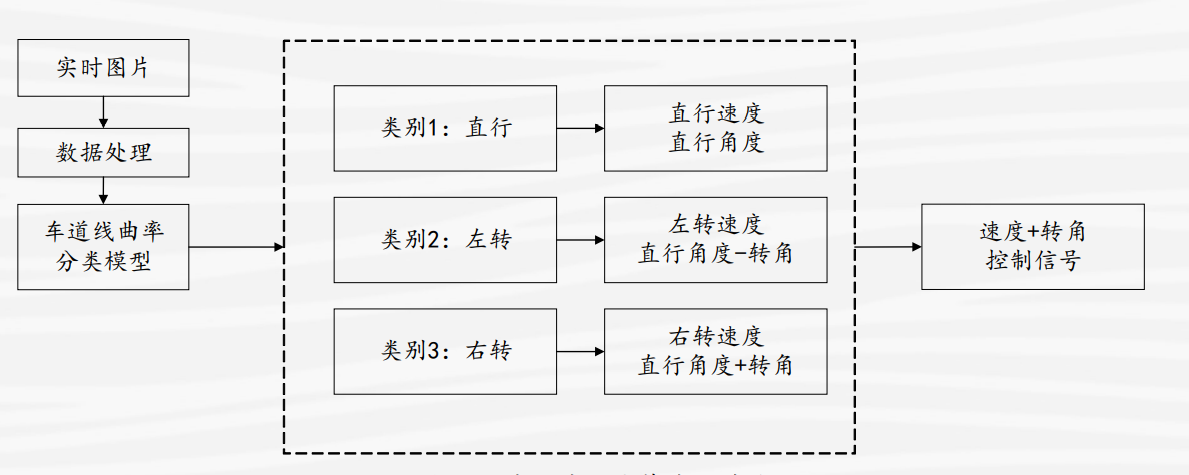
3**.4 数据处理及模型训练**

模型检测集成到智能车上需要如图 3-6 所示的两大阶段，分别是数据处 理和模型构建。其中在数据处理阶段主要包括数据的采集、标注、划分及数据增广，模型构建部分主要包括目标检测模型的训练、导出以及在车端的部署。

**3.5 循迹算法**

循迹算法要解决的核心问题为：在仅考虑车道线识别结果，而不 考虑目标检测和障碍物检测结果的情况下，如何根据道路类别设置转角信息， 从而使智能车的性能保持在最佳状态。

根据核心问题可知，循迹算法将涉及对速度和转角的适配问题，当速度慢的时候，除直线外两种类别的转角可以设置较小，因为低速不至于冲出赛道，且小转角有利于智能车的平稳行驶；当速度快的时候，除直线外的两种类别的转角应该设置偏大，否则在弯道时智能车来不及转正 姿态的情况下可能就已经驶离赛道。



因此，对于端到端的循迹算法的设计，应该设置好参数接口以便于在大量的实 车调试过程中对参数进行调整。

不同于例程中给定的控制流程，本参赛队将此部分的算法优化成下述流程：

1. 初始化列表，每个列表元素为类别、对应速度和转角；
2. 加载道路分类；
3. 实时从相机中读取图像数据，并将数据处理成道路分类器的输入数

据

1. 使用道路曲率分类器对输入图片进行推理；

(5) 根据推理结果匹配(1)中对应的速度和角度构成控制信号下发给控制器。

1. **关键算法实现**

**4.1 数据集制作相关算法**

**4.1.1 数据采集**

模型采集采用一个采集算法。该采集算法在 例程的基础上进行优化，增加断点存储、图片删除、文件夹更换等功能。

**4.1.2 数据集制作**

使用自制标注软件对采集到的图像进行标注，设置图像的存储目录和标签的存 储目录后，开始进行标注，直到文件夹标注完成后继续下一个文件夹的标注。

在得到原始图片和对图片的标注后，需对其划分成训练集和验证集，用于目标 第20页 检测模型的训练。首先将标号和对应的标签写入 label 文件，然后将图片的路径和从 该图片中解析出的标签信息写入到用于训练的train.txt和用于验证的eval.txt文件中。 此处，训练集和验证集划分的比例为 8：2。

1. **测试**

**5.1 测试环境说明**

在各个阶段开发完成后，需要对该阶段进行单元测试和集成测试，单元测试由 开发人员在软件平台上完成，集成测试则由小组测试人员移植到车端后完成。因此， 在这里对测试的硬、软件环境和集成测试的场地环境进行简要说明。

1. 硬件平台

车端平台主要使用组委会统一提供的 I 型车模，其基本参数列举如下：  ①底盘：四轮带编码器差速，尺寸为长340mm× 宽270mm× 高 300mm

②电机：G37-520B 编码器直流电机，12V，空载转速为 178rpm；

③摄像头：720P 像素，对角 70°，水平 55°，YUY2/10 至 15 帧/s；

④处理器：百度大脑 EdgeBoard 深度学习计算卡；

其余还包括开关、数码管电量显示、遥控器、电池、充电器等硬件组件。

对于道路曲率分类模型和标志物检测模型的训练，均使用百度提供的深度学习 一体化学习与实训平添 AI Studio 提供的环境，该环境的硬件配置如下： 

CPU：Intel（R）Xeon（R）Gold 6148，内存为 32GB； 

GPU：NVDIA Tesla V100，显存为 16GB； 

磁盘空间：100GB；

1. 软件平台

车端处理器 EdgeBoard 软件环境配置如下： 

操作系统：Linux 内核，RootFS； 

编程语言：C++。

深度学习框架：PadddlePaddle Lite；

AI Studio 环境的软件配置如下： 

操作系统：Ubuntu 16.04 LTS； 

编程语言：Python 3.7； 

框架版本：PaddlePaddle 1.7.1；

1. **小车总结**

**6.1 技术创新点**

在技术创意方面，本作品在技术路线中各个模块的数据采集、数据处理、模型 训练等环节，以及最后的系统集成环节都有不同程度的创新，其中最有效的创新是 技术路线中的端到端循迹模块的设计与实现，以及系统集成环节中状态机的设计与实现。

**6.2 技术路线**

整车系统中的决策与控制中心实时汇总三大模块输出的道路类型信息、沿途模型类别与位置信息，以及道路障碍物检测信息，给出当前态势下车体需要执行的线速度与角速度命令，车体底盘接收到控制命令后，控制电机进行相应动作，带动车体按规定运动，最终完成指定任务。经过各项功能测试表明，该技术路线体系完整，且运作有效，可以很好的完成本次比赛规定的各项任务。

**6.3 存在的问题及改进方向**

回顾小车技术方案从设计到实现的全过程，整理总结出以下几点目前存在的 问题，并提出相应的改进方向：

1. 循迹的轨迹不平滑。产生这个 问题的主要原因是因为本作品的循迹模块是基于三分类模型设计的，该模型将道路曲率简单的划分为直行、左转和右转，这样划分太过粗糙，不能对直道或弯道中小曲率路段的行驶做到精细控制。解决该问题可以继续采用分类的思想，进一步增加道路类型的类别 数目，用离散逼近连续。
2. 目标检测结果误检率很高。如果没有引入状态机，基本无法完成比赛任务。 产生这个问题的主要原因是因为在进行模型训练时，其输入必须是包含目标的图片，而不能包含完全没有目标标志物的背景图片，从而导致网络训练时没有得到足够的负样本，进而在实车测试时，在无目标标志物的路段经常发生误检。下一步工作可以对官方给出的的模型与训练代码进行进一步的优化，使其可以接受无目标的背景图片作为训练数据。