本文引用格式：杨静静,曹海平,丁杰,等.基于 EdgeBoard 的智能循迹避障小车设计[J].自动化与信息工程,2024,45(2):22-28.
YANG Jingjing, CAO Haiping, DING Jie, et al. Design of intelligent tracking and obstacle avoidance vehicle based on EdgeBoard[J]. Automation & Information Engineering, 2024,45(2):22-28.

基于 EdgeBoard 的智能循迹避障小车设计*

杨静静　曹海平　丁杰　顾银波　霍旋
（南通大学张謇学院，江苏 南通 226000）

摘要：针对传统的自主循迹避障小车在复杂路况下存在图像识别精度低、处理时间长等问题，设计一种基于 EdgeBoard 的智能循迹避障小车。首先，通过 USB 摄像头采集道路图像；然后，采用基于 OpenCV 的道路循迹算法识别道路边缘；最后，利用 SSD_MobileNetV1 深度学习模型识别特定的交通标志。经测试，该智能循迹避障小车具有良好的稳定性和可靠性。

关键词：循迹避障小车；EdgeBoard；OpenCV；深度学习；自主循迹避障

中图分类号：TP242.6　　**文献标志码：**A　　**文章编号：**1674-2605(2024)02-0004-07

DOI：10.3969/j.issn.1674-2605.2024.02.004

Design of Intelligent Tracking and Obstacle Avoidance Vehicle Based on EdgeBoard

YANG Jingjing　CAO Haiping　DING Jie　GU Yinbo　HUO Xuan
(Zhangjian School, Nantong University, Nantong 226000, China)

Abstract: Aiming at the problems of low image recognition accuracy and long processing time in traditional autonomous obstacle avoidance vehicles under complex road conditions, a smart obstacle avoidance vehicle based on EdgeBoard is designed. Firstly, collect road images through a USB camera; Then, the road tracking algorithm based on OpenCV is used to identify the edges of the road; Finally, use the SSD_MobileNetV1 deep learning model to identify specific traffic signs. After testing, the intelligent tracking and obstacle avoidance car has good stability and reliability.

Keywords: tracking and obstacle avoidance vehicle; EdgeBoard; OpenCV; deep learning; autonomous tracking and obstacle avoidance

0　引言

随着社会经济的发展，汽车数量不断攀升，给人们生活带来便利的同时，也带来交通拥堵和环境污染等问题。自动驾驶技术以摄像头、雷达、超声波、传感器等感知器件和图像处理、神经网络控制等技术为基础，实现汽车之间的智能协同和路径规划，可有效提高行驶效率，缓解交通压力[1]。

自主循迹避障是自动驾驶技术的重要组成部分。目前，自主循迹避障技术已经取得较大进展。文献[2]提出一种红外循迹的设计方案，但对跑道宽度要求较高。文献[3]采用混合遗传算法设计智能循迹小车，但受电磁干扰较强。在复杂场合下，如何合理地规划路径是当前自主循迹避障研究领域关注的重点。

为此，本文提出一种基于 EdgeBoard 的智能循迹避障小车（以下简称“智能车”），采用基于 OpenCV 的道路循迹算法和 SSD_MobileNetV1 深度学习模型，实现在不同道路条件下的稳定行驶和及时避障，提升了智能车的安全性能和智能水平。

1　智能车组成架构

本文设计的智能车主要包含视觉感知模块、路径规划决策模块、运动控制模块等，总体架构如图 1 所

* 基金项目：江苏省高等学校省级创新创业训练计划项目（202210304204H）

示[4]。其中，视觉感知模块通过摄像头获取智能车周围的道路信息；路径规划决策模块的控制器上位机结合运动控制核心下位机，规划决策智能车的行驶路径，由此控制并驱动运动控制模块中的电机和舵机，实现智能车循迹避障。

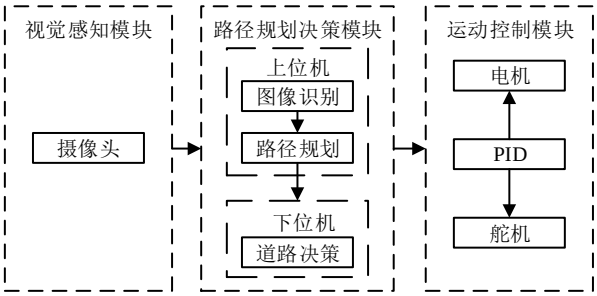


图 1　智能车总体架构图

2　硬件设计

智能车的视觉感知模块选用 USB 摄像头，上位机采用百度配置的 EdgeBoard，运动控制模块选用 GA12-N20 直流电机和 MG995 舵机；车身选用 I 型车模，采用碳纤维底盘搭建。

智能车的下位机主要包括核心板、驱动板、主控板。其中，核心板是下位机的控制中枢，用于编程开发和运动控制；驱动板接收核心板的控制指令，以驱动电机、舵机转动，二者均采用 TC264 系列芯片；主控板根据智能车的需求自主设计，用于智能车供电，其部分电路图如图 2 所示。

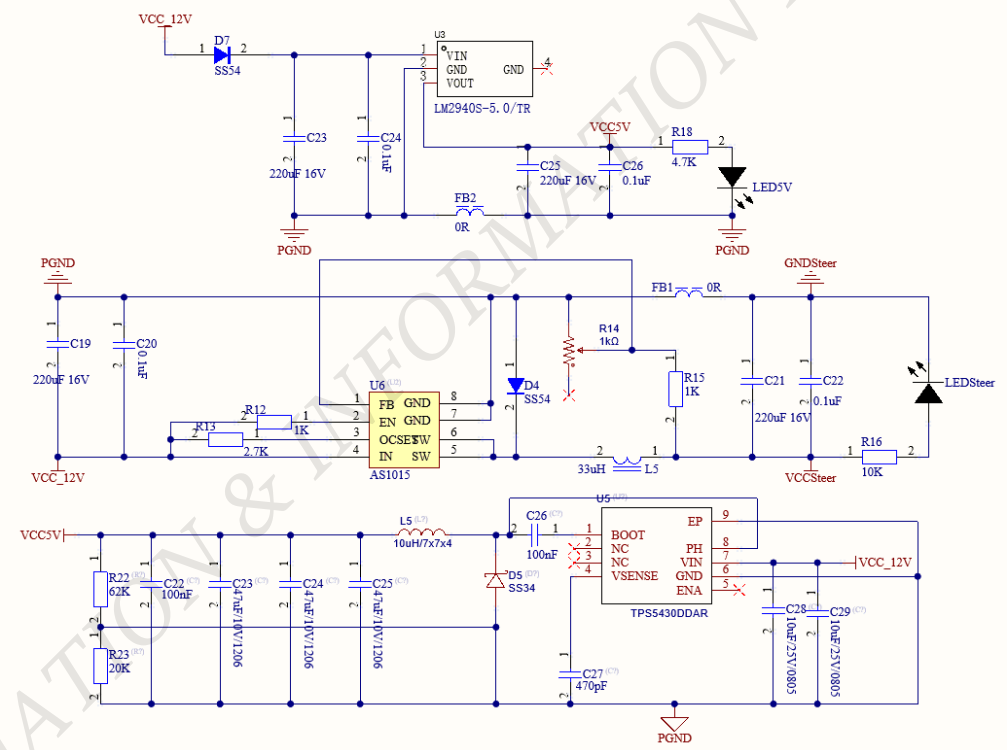


图 2　智能车主控板部分电路图

智能车由 5 V 电源、3.3 V 电源、11.1 V 锂电池 3 种电源供电。其中，5 V 电源选用线性稳压芯片 TPS5430，为电机供电；3.3 V 电源选用线性稳压芯片 AS1015，为单片机系统、舵机供电；11.1 V 锂电池为下位机的电路板供电，电路板通过 LM2940S 变压器将 11.1 V 电压供给电机、舵机等设备。此外，该电路还预留了引脚，用于连接驱动板和舵机，通过上位机与下位机的联合控制，使驱动板和舵机分别实现智能车的速度、角度控制[5]。

3　智能车控制流程

智能车控制流程如图 3 所示。

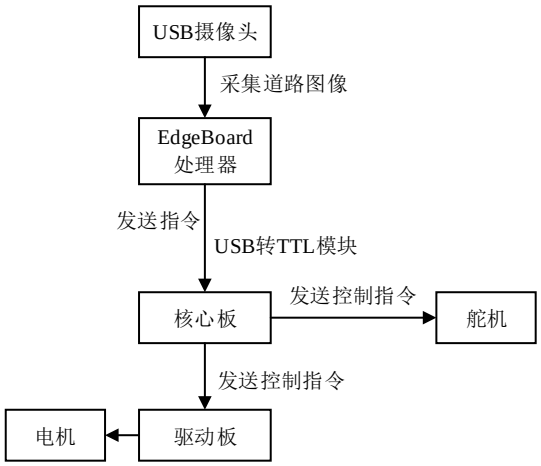


图 3　智能车控制流程

通过安装在智能车上的 USB 摄像头采集道路图像；利用基于 OpenCV 的道路循迹算法及 SSD_MobileNetV1 深度学习模型，提取道路图像的道路边缘信息和图像特征，进而判断道路类型；EdgeBoard 处理器通过 USB 转 TTL 模块发送加密指令给下位机；下位机解析收到的指令，并发送控制指令给电机、舵机等设备；电机采用 PID 控制算法，通过 PWM 控制驱动电路调整转速，实现电机转速的闭环控制，进而实现智能车的速度控制；舵机根据接收的 PWM 信号转动对应的角度，实现智能车的角度控制。

4　软件设计

4.1　总体方案

智能车的软件采用 SSH 客户端工具 FinalShell 进行开发和调试。FinalShell 提供了可视化的控制界面，方便对 EdgeBoard 进行源程序编写、编译和链接等操作，并生成可执行文件。

智能车的软件主要包括基于 OpenCV 的道路循迹算法和 SSD_MobileNetV1 深度学习模型。其中，基于 OpenCV 的道路循迹算法可实现图像预处理、二值化、道路识别等功能；SSD_MobileNetV1 深度学习模型对 USB 摄像头采集的特定交通标志进行有效标注，并利用神经网络进行模型训练和部署[6]。智能车的软件设计流程图如图 4 所示。

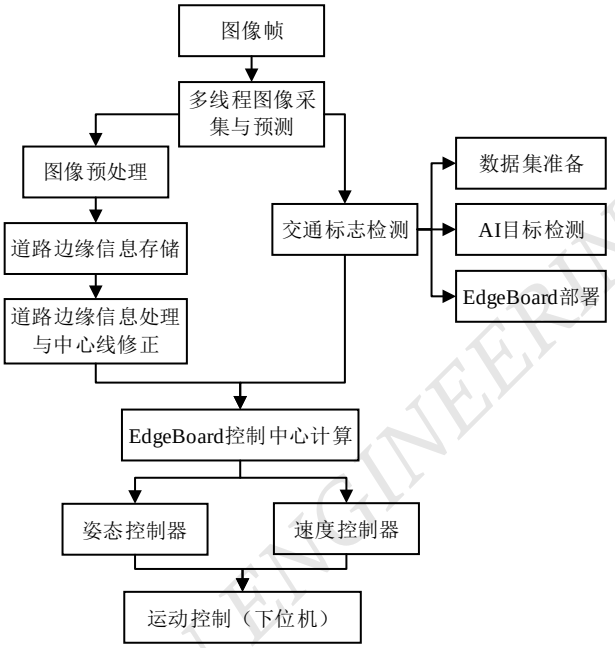


图 4　智能车软件设计流程图

4.2　基于 OpenCV 的道路循迹算法

4.2.1　图像预处理

首先，利用颜色空间转换函数 cvtcolor()对 USB 摄像头采集的道路图像进行灰度化处理[7]，简化图像信息，突出图像特征，提高后续的处理效率。

然后，利用（3,3）卷积核的高斯滤波对灰度化处理后的道路图像进行平滑处理，更多地保留道路图像的总体灰度分布情况。

接着，利用 OSTU 算法进行二值化处理。通过最大化背景和道路之间的类间方差来区分背景和道路。类间方差越大，背景和道路的灰度差异越明显，错分的可能性越小[8]。

最后，得到由 255（白色）和 0（黑色）组成的道路预处理图像，如图 5 所示。

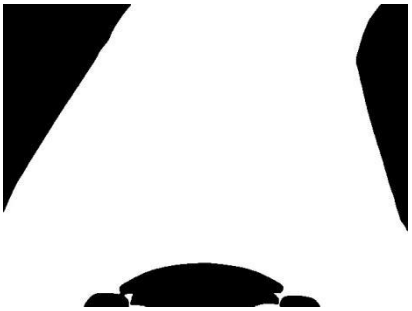


图 5　道路预处理后的图像

4.2.2 道路边缘信息存储

在预处理后的道路图像中，对每行色块进行搜索，找出相邻两列之间的突变色块。按照行列大小关系，分别存储左右突变色块的信息。左右突变色块搜索存储的主要代码为：

```
if (imagePath.at<uchar>(row, col) > 127 &&
   imagePath.at<uchar>(row, col - 1) <= 127)
{
  startBlock[counterBlock] = col;   //黑色 0,0,0，每行白色起点列数存储
}
else
    if (imagePath.at<uchar>(row，col) <= 127 &&
      imagePath.at<uchar>(row, col - 1) > 127)
    {
       endBlock[counterBlock++] = col;     //每行白色终点列数存储
       if (counterBlock >= end(endBlock) - begin( endBlock) )
         break;
    }
```

如果某行的色块宽度大于设置的色块宽度 widthBlocks，则认为其是左边缘或右边缘的信息，分别存储到 pointsEdgeLeft 和 pointsEdgeRight 两个 vector 向量中。

4.2.3 道路边缘信息处理与中心线修正

根据实时存储的道路边缘信息，搜索边缘信息中突变行的左上（rowBreakLeftUp）、左下（rowBreakLeftDown）、右上（rowBreakRightUp）、右下（rowBreakRightDown）4 个点，以便后续进行道路边缘信息的更新[9]。

以搜索十字道路的左上边缘突变点为例，从左边缘信息的末尾元素位置开始向上搜索，判断当前点的纵坐标是否大于 2，并且与下一个点的纵坐标差是否小于 3。如果满足条件，说明当前点不是突变点，更新 rowBreakLeftUp 的值为当前索引 i，继续向上搜索；同时重置突变点计数器 counter 为 0，增加过滤计数器 counterFilter 值。

若 counterFilter 的值大于 10，且当前点的纵坐标突然小于等于 2，表示可能找到了潜在的突变点，counter 的值增加 1。如果 counter 累计值超过 5，表示找到了连续的左上边缘突变点，正式返回 rowBreakLeftUp 的值（以上参数数值根据实际情况均可调）。

搜索十字道路左上边缘突变点的主要代码片段：

```
for (int i = pointsEdgeLeft.size() - 5; i > 50; i--)
        {
            if (pointsEdgeLeft[i].y > 2 &&
abs(pointsEdgeLeft[i].y - pointsEdgeLeft[i + 1].y) < 3)
            {
                rowBreakLeftUp = i;
                counter = 0;
                counterFilter++;
            }
            else if (pointsEdgeLeft[i].y <=2 &&
counterFilter > 10)
            {
                counter++;
                if (counter > 5)
                    return rowBreakLeftUp;
            }
        }
```

搜索结束后，由左上、左下、右上、右下 4 个突变点求取 2 条直线。设一直线 L 上的两点 P_1、P_2 的坐标分别为 (x_1, y_1)、(x_2, y_2)，且 $(x_1 \neq x_2)$，则直线 L 的斜率、截距 b 分别为

$$\begin{aligned} k &= (y_2 - y_1) / (x_2 - x_1) \\ b &= y_1 - k \bullet x_1 \end{aligned} \tag{1}$$

将边缘突变行的左上、左下、右上、右下 4 个点分别代入公式(1)，计算出 2 条直线的斜率、截距，可得到 2 条直线方程。将位于左上到左下的行值分别代入 2 条直线方程中的 x，求出对应的 y 值，即新的左右边缘点列值。更换原来的边缘点列值，并存入边缘点集中，道路边缘信息更新完成。

在更新后的道路左右边缘信息中，首先，分别选取图像的首行边缘点、1/3 处边缘点、1/2 处边缘点、末行边缘点，共 4 对左右边缘点；然后，将 4 对左右边缘点的行值、列值之和分别除以 2，得到 4 对边缘点的中点；最后，由 4 对边缘点的中点坐标生成三次贝塞尔曲线：

$$P(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3, t \in [0,1] \tag{2}$$

式中：P_0、P_1、P_2、P_3为贝塞尔曲线的控制点，$P(t)$为贝塞尔曲线上参数t处的点坐标[10]。

将 4 对边缘点的中点坐标代入公式(2)，得到的曲线即为道路的中心线。通过对比道路中心线与道路图像中心的偏差及方向，计算需要修正的角度，计算方法如下：

```
float error = controlcenter - cOLSIMAGE / 2; //图像控制中心转换偏差
static int errorLast - e;//记录前一次的偏差
if (abs(error - errorLast) > cOLSIMAGE / 10)
{
    error = error > errorLast ? errorLast + COLSIMAGE / 10 :
    errorLast - coLSIMAGE / 10;
}
```

根据 PID 控制算法的实时反馈，实现智能车方向和速度的精准控制，使其沿着道路中心线行驶。

4.3 SSD_MobileNetV1 深度学习模型

智能车需要识别特定的道路交通标志，以完成特殊路段的通行。本文通过 SSD_MobileNetV1 深度学习模型识别特定的道路交通标志。

4.3.1 数据集的采集与制作

将安装了 USB 摄像头的智能车放置在采集点，在 EdgeBoard 上运行 collect.py 图像采集程序，通过手柄控制智能车前进，采集各路段的道路图像。

利用开源软件 LabelImg 对采集的道路图像进行标注，将生成的 xml 与道路图像一一对应，打包成数据集。将标注好的数据集的 70%划分为训练集、15%划分为验证集、15%划分为测试集。

4.3.2 模型训练

由于 EdgeBoard 处理器的计算资源有限，智能车选取 SSD_MobileNetV1 网络进行模型训练，以免出现帧率较低的情况。

SSD_MobileNetV1 采用 MobileNetV1 作为主干特征提取网络，共使用 5 个不同尺度的特征图，预测不同大小比例的回归候选框，网络结构如图 6 所示。

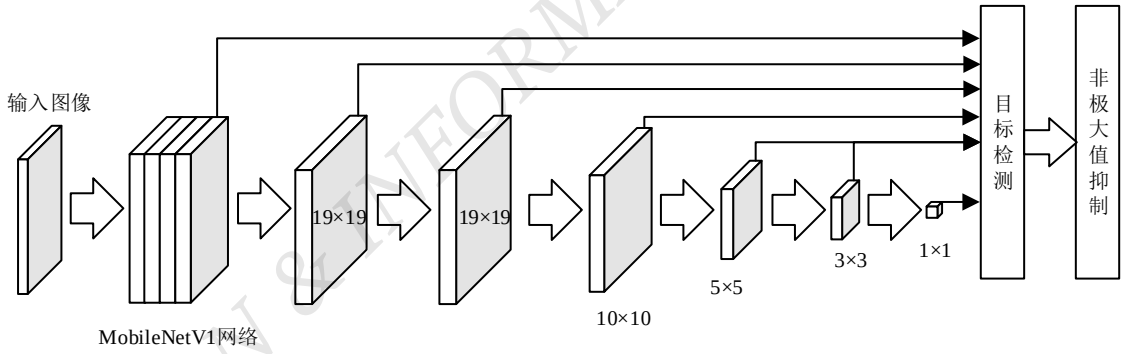


图 6　SSD_MobileNetV1 网络结构

利用 SSD_MobileNetV1 模型识别特定的道路交通标志时：

首先，对输入的道路图像进行深度可分离卷积和池化操作，得到 6 个不同尺寸和深度的特征图；

然后，通过多层卷积和池化操作，设置锚框（8 732 个），得到一系列预测框和对应的置信度分数，通过锚框的大小和长宽比确定特征图层数和特定长宽比的比例尺；

接着，利用非极大值抑制法，选择一组置信度高、不重叠的预测框，作为初始目标的检测结果；

最后，计算高置信度分数的预测框与真实目标框之间的 IoU 值，并进行比对，以确定每个预测框是否包含目标，将这些预测框进行分类标记并计算预测框的位置回归，得到最终目标的检测结果。

此外，本文采用综合考虑分类误差和边界框误差的 MultiBox 损失函数，并通过反向传播算法更新模型参数，使模型的性能和泛化能力进一步提高。

利用训练后的 SSD_MobileNetV1 模型识别特定的道路交通标志，识别效果如图 7 所示。

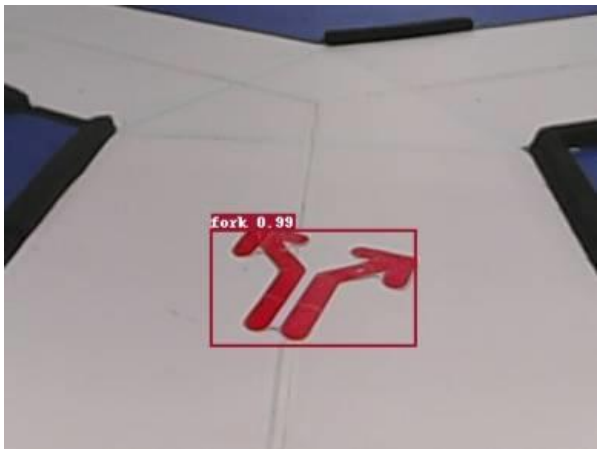


图 7　特定的道路交通标志识别效果

由图 7 可知，SSD_MobileNetV1 模型可识别 USB 摄像头拍摄的交通标志和对应概率。

5　智能车测试

为测试智能车在不同光照条件下的稳定性和性能，在正常光照（12 时，室内未开灯）和较暗光照（18 时，室内开灯）下分别进行 15 次测试。

测试道路情况：地图长为 5.58 m，宽为 5.21 m，道路总长度为 27.9 m。

测试过程：

1） 准备工作，确保智能车和测试环境处于正常状态，包括智能车的电池电量稳定在 75%以上、USB 摄像头正常工作并提供清晰、稳定的图像；

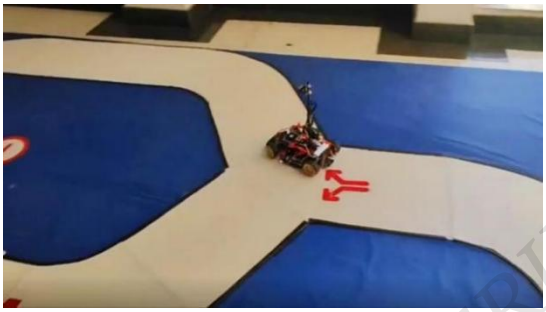
2） 开始测试，在正常光照/较暗光照条件下，将智能车放置在车库位置，秒表归零，启动智能车主程序，开始记录时间；

3） 测试过程，记录开始进入设置路段和完全通过设置路段的时间点；

4） 结束测试，当智能车第二次抵达车库并完全进入车库时（如图 8(c)所示），记录此时的时间作为计时终点；

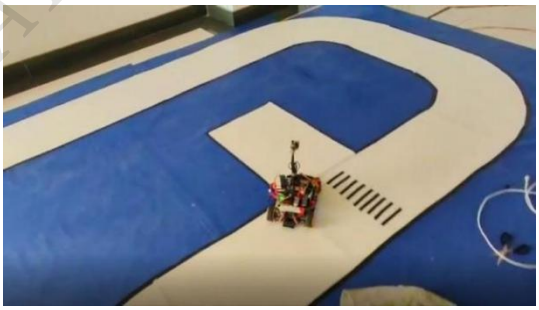
5）多次测试，为了确保测试结果的可靠性和稳定性，进行多次测试。

测试效果如图 8 所示，通过时间如表 1 所示。



(a)　智能车通过三岔路口

(b)　智能车通过十字路口



(c)　智能车入库

图 8　智能车测试效果图

表 1　智能车 30 次测试通过时间　　单位：s

设置路段	测试环境	
	正常光照	较暗光照
车库	0.97	1.02
三岔路口	3.45	3.77
十字路口	2.98	3.20
避障区域（塑料锥桶放置区域）	6.08	6.47

由表 1 可知，在正常光照与较暗光照条件下，智能车各设置路段的通过时间相差在 0.5 s 内，可以顺

利完成出入库、三岔拐弯等任务，具有良好的稳定性和可靠性。

6 结论

本文设计了一种基于EdgeBoard的智能循迹避障小车。测试结果表明；该智能车可以在设定的道路上实现自主规划、识别特定的交通标志等功能，具有良好的稳定性和可靠性。但智能车运行算法逻辑较复杂，代码运行效率较低，后续会对智能车的运行算法进行优化，减少代码冗余部分，提高运行效率。

参考文献

[1] 曹月花,李辉.一种基于 EdgeBoard 的智能车系统设计与实现[J].现代电子技术,2022,45(18):166-170.

[2] 张萍,陈国壮,候云雷,等.模糊控制红外循迹小车的研究[J].实验室研究与探索,2018,37(7):50-53;91.

[3] 莫太平,周园园,张云强.面向工业物流服务的智能车及循迹算法的研究[J].火力与指挥控制,2017,42(3):146-151.

[4] 张子涵.基于多传感数据融合的自主巡航智能小车设计[D].杭州:浙江理工大学,2023.

[5] 黄凯龙.基于飞思卡尔单片机的智能车及其调试系统设计[D].长沙:湖南大学,2014.

[6] 于亚威.深度学习条件下的多摄像机行人视频目标再识别研究[D].深圳:深圳大学,2019.

[7] 雷得超,任守华.基于 OpenCV 图像处理车牌识别系统分析研究[J].电脑与信息技术,2022,30(4):15-17.

[8] 蔡文杰,丁青.基于大津算法的 5GNR 帧结构自动检测方法[J].通信技术,2021,54(2):369-373.

[9] 杨萍,侯静茹,曹强.基于单片机的智能车图像处理与道路识别算法研究[J].机械制造,2017,55(1):32-35.

[10] 江浩,彭侠夫.基于三次贝塞尔曲线的轨迹规划方法[J].数字技术与应用,2022,40(11):7-10.

作者简介：

杨静静，女，2002 年生，本科，学生，主要研究方向：电气工程及其自动化。

曹海平（通信作者），男，1972 年生，硕士研究生，高级实验师，主要研究方向：电气控制。E-mail: cao.hp@ntu.edu.cn

丁杰，男，2002 年生，本科，学生，主要研究方向：电气工程及其自动化。

顾银波，男，2002 年生，本科，学生，主要研究方向：计算机科学与技术。

霍旋，男，2002 年生，本科，学生，主要研究方向：通信工程。

（上接第 6 页）

参考文献

[1] 王新扬.智能网联汽车技术发展现状刍议[J].汽车维修技师,2023(3):123-124.

[2] 方敏,张立新,于星胜.浅谈智能网联汽车的发展[J].内燃机与配件,2020(14):184-185.

[3] 李克强.我看智能网联汽车十年发展[J].智能网联汽车,2022(3):6-9.

[4] 刘岩.智能网联汽车发展现状及趋势[J].河北农机,2021(11):62-63.

[5] 马兰兰.智能网联汽车业务发展战略研究[J].专用汽车,2022(10):6-8. DOI:10.19999/j.cnki.1004-0226.2022.10.003.

[6] 工业和信息化部印发《车联网(智能网联汽车)产业发展行动计划》[J].智能制造,2019(Z1):6.

[7] 国家发展改革委等十一部门联合印发《智能汽车创新发展战略》[J].环境技术,2021,39(1):4.

[8] 广东省工业和信息化厅 广东省公安厅 广东省交通运输厅关于印发广东省智能网联汽车道路测试与示范应用管理办法(试行)的通知[Z].广东省人民政府公报,2022(35):17-25.

[9] 深圳市交通运输局 深圳市发展和改革委员会 深圳市工业和信息化局 深圳市公安局交通警察局关于印发《深圳市关于推进智能网联汽车应用示范的指导意见》的通知[Z].深圳市人民政府公报,2020(35):20-23.

作者简介：

尚学峰，男，1989 年生，硕士研究生，副研究员，主要研究方向：科技情报、产业技术发展战略、新能源汽车产业技术发展。E-mail: 924151224@qq.com