

Vehicle Lane Assist System

Vanshika Singh; Nikita Tiwari; Rayyan Anshari; Shahnawaz Shaikh; Mahesh Kadam (Prof.)
Department of Electronics & Telecommunication Engineering
Terna Engineering College, Navi Mumbai

Abstract:- This research paper presents the development of a fast and reliable lane detection device designed to detect and alert the user for efficient and safe journey. Our lane detection device makes it easier to see lane markings on the road and warns the user when he switches to wrong side of road. This budget friendly device will provide the real time data with the help of sensors and used it for google maps integration. Our device is even flexible for extreme weather conditions.

Keywords:- Lane Detection , Sensors, Google Maps, Alert.

I. INTRODUCTION

Road safety continues to be a major developmental issue, and public health concern, and according to 'ministry of Road Transport & Highway' road accidents are one of the leading causes of death, disability, and hospitalization of people worldwide in general and Indian particular. At least one out of 10 people killed on roads across the world is from India. With the growth in traffic, road safety becomes more and more essential. Most accidents on the roads are caused by drivers leaving the lane without obeying the correct laws. Dealing with these issues our project deals with distracted driving when the driver diverts his/her attention from the road. Hence, we are introducing lane detectors that will detect the lane with the help of sensors and alert the driver by ringing the alarm. The combined output of sensors will provide better safety for road transport. Using Image processing we are providing more data to our system so that our sensor can detect lane/road precisely. Our lane detecting device makes it easier to see the lane markings on the road and warns the user when he switches to the wrong side of road. In case of broken white lines, the alarm will not generate, and it will allow the vehicle to overtake. In lane detection for vehicles a lot of technologies are used for edge detection and alert if the vehicles go out of the lane. Mainly it is done using OpenCV using computer vision.

Received data will integrate the data sources into a single web application to provide onsite service. In this way, the data are accurate and up to date due to the maintaining of those agencies. Our lane detector is budget friendly and can be useful for all cars, and trucks for all price ranges.

II. BLOCK DIAGRAM

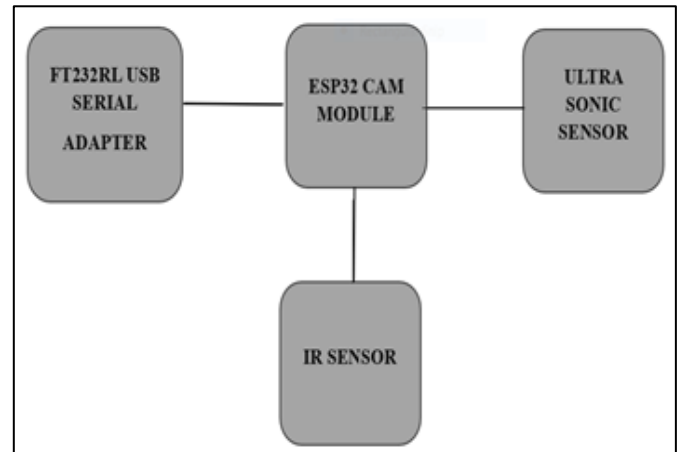


Fig. 1 Block Diagram

Fig.1 the block diagram outlines a system for lane detection and object recognition in a vehicle using computer vision and sensor fusion. The system starts by capturing data from an ESP32 camera and ultrasonic and IR sensors. The lane detection block utilizes image processing techniques to analyze the camera footage and identify lane markings on the road.

In parallel, the ultrasonic and IR sensor data is processed to detect objects around the vehicle. This sensor fusion block combines the lane detection results with the object detection information to create a comprehensive picture of the surrounding environment.

Optionally, the system can integrate with Google Maps API to potentially improve lane detection in situations with poor visibility (e.g. fog). Finally, all processed data is displayed on a screen, likely showing the lane markings, detected objects, and other relevant information.

III. HARDWARE REQUIREMENT

➤ ESP32 CAM Module



Fig 2 ESP32 CAM Module

The ESP32-CAM is a compact development board based on the ESP32 chip, which integrates a Wi-Fi and Bluetooth module along with a camera. It is commonly used for IoT projects, home automation, and surveillance applications due to its small size, low power consumption, and versatile capabilities. However, it's important to note that while the ESP32-CAM can capture images and video, its default configuration does not include advanced image processing or lane detection algorithms. For lane detection specifically, you would typically need to implement additional software and algorithms on top of the ESP32-CAM's capabilities. This would involve tasks such as capturing images or video frames from the camera, processing them to detect lane markings or edges, and then applying decision-making logic to determine the vehicle's position within the lane.

➤ Ultrasonic Sensor



Fig 3 Ultrasonic Sensor

An ultrasonic sensor is a device that utilizes sound waves with frequencies higher than the human audible range to measure distances to objects. These sensors are widely used in various applications such as robotics, automotive systems, industrial automation, and proximity sensing due to their accuracy, reliability, and ease of use. However, it's important to note that ultrasonic sensors alone may not be sufficient for accurate lane detection, especially in complex scenarios. They are typically used as part of a sensor fusion system that combines data from multiple sensors to determine lane position and detect obstacles.

➤ FT232RL USB TTL Serial Adaptor

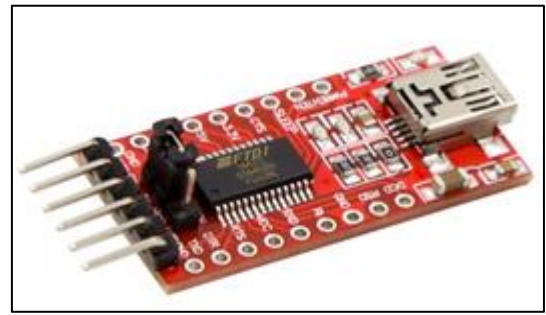


Fig 4 FT232RL USB TTL Serial Adaptor

The FT232RL USB to TTL Serial Adapter is a popular integrated circuit (IC) manufactured by FTDI (Future Technology Devices International) that provides a convenient way to interface serial devices with USB-equipped computers or microcontrollers. The FT232RL USB TTL Serial Adapter is a versatile component commonly used in electronics projects for communication between devices with UART (Universal Asynchronous Receiver-Transmitter) serial interfaces and computers or microcontrollers via USB. However, it's important to note that the FT232RL adapter itself does not directly contribute to lane detection. Instead, it serves as a communication bridge and interface component in the overall system. FT232RL USB TTL Serial Adapter plays a crucial role in facilitating data transfer and communication within a lane detection system, its direct contribution to lane detection lies in enabling the seamless integration and communication between various components such as sensors, microcontrollers, and computers for effective data processing and analysis.

IV. SOFTWARE REQUIREMENT

- Frontend (JavaScript - MERN Stack):
- React.js: A JavaScript library for building user interfaces. You can use React for creating interactive components and managing the frontend of your web application.
- React Router: A routing library for React applications. It allows you to handle navigation and define routes within your React application.
- Material-UI: A popular React component library that provides pre-designed UI components following the Material Design guidelines. It can help you quickly build a visually appealing frontend for your application.
- Axios: A promise-based HTTP client for making asynchronous HTTP requests in JavaScript. You can use Axios to communicate with your backend API from the frontend.
- Mapbox or Leaflet: JavaScript libraries for working with maps and geospatial data. You can use one of these libraries to display maps and integrate features like markers and overlays into your frontend application.
- Backend (Python):
- Flask or FastAPI: Lightweight Python web frameworks for building backend APIs. You can use Flask or FastAPI to create RESTful endpoints for handling requests from your frontend application.

- **OpenCV:** An open-source computer vision library for Python. OpenCV provides a wide range of functions and algorithms for tasks such as lane detection, object detection, and image processing.
- **NumPy:** A fundamental package for scientific computing with Python. NumPy provides support for multi-dimensional arrays and matrices, which are essential for handling image data in OpenCV.
- **MongoDB:** A NoSQL database that stores data in JSON-like documents. You can use MongoDB to store and manage data collected from your application, such as user preferences or detected objects.
- **Pymongo:** A Python driver for MongoDB. Pymongo allows you to interact with MongoDB databases from your Python backend, including performing CRUD (Create, Read, Update, Delete) operations and querying data.
- **Requests:** An HTTP library for Python. You can use Requests to make HTTP requests to external APIs, such as the Google Maps API, from your Python backend.
- **Google Maps API:** A set of APIs provided by Google for working with geographic data and integrating maps into web applications. You can use the Google Maps API to display maps, geocode addresses, calculate directions, and more.

V. WORKING AND IMPLEMENTATION

Our backend infrastructure for the lane detection system is now fully set up, integrating essential Python libraries and technologies to ensure efficient functionality and enhance road safety. Firstly, we installed Flask using pip as our chosen backend framework, allowing us to handle HTTP requests and define API routes for various functionalities. This setup includes endpoints for image processing using OpenCV and NumPy, which enable tasks such as object detection and, more crucially, lane detection. The integration of the Ultrasonic sensor further enhances our application's capabilities by providing distance measurement or object detection functionalities.

In terms of data storage and management, we installed MongoDB and utilized PyMongo to interact with MongoDB databases efficiently. This allowed us to implement CRUD operations for managing data collected from the application, including storing processed images and user preferences. A well-designed data schema within MongoDB ensures the efficient storage and retrieval of processed data or results from image processing tasks, contributing to the system's reliability.

Additionally, we integrated the Requests library to facilitate HTTP requests to external APIs, specifically integrating the Google Maps API. This integration enhances the lane detection system's accuracy and reliability by providing real-time data, geocoding addresses, calculating directions, and displaying maps. Security considerations were also a top priority, as we implemented necessary measures such as authentication, authorization, and data validation to ensure a secure backend environment and protect sensitive information and API keys during interactions with external APIs.

Lastly, thorough testing of the backend APIs was conducted to ensure functionality and performance across different endpoints, including image processing and external API interactions. The backend setup is now ready for deployment on hosting platforms such as Heroku, AWS, or DigitalOcean, making it accessible to frontend applications and end-users. Overall, our backend infrastructure leverages a combination of Flask, OpenCV, NumPy, MongoDB, PyMongo, Requests, and the Google Maps API, playing a crucial role in providing real-time alerts and enhancing road safety for drivers.

VI. CONCLUSION & FUTURE SCOPE

The project outlined in this document introduces a system with the capacity to fuse camera sensor data with the Google Maps API, offering a platform for deriving profound insights from the real world.

The system detailed in this document has the potential to be a valuable tool for extracting meaningful insights from the real world, improving our understanding of the world around us, and driving progress across various domains. It embodies the essence of innovation, adaptability, and the pursuit of a safer and more informed future.

In future it can be used for important driving accessories that can even store the data for better predications in situation.

REFERENCES

- [1]. J. Murphy, Y. Gitman. Pulse Sensor Advanced”<https://pulsesensor.com/pages/code-and-guide/>[retrived June 2020]
- [2]. Kahtam A., Saed T., Salah H. Smart real-time healthcare monitoring and tracking system using GSM/GPS technologies. Big Data and Smart City (ICBDSC), 2017 3rd MEC International Conference on. 28 April 2018. Muscat, Oman
- [3]. David Thompson, “Raspberry Pi 3 Model B”<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>[retrived June 2019]
- [4]. Deffensive POW/MIA Accounting Agency. Soldier Missing From Korean War Accounted for
- [5]. Hunt.<http://www.dpaa.mil/News-Stories/NewsReleases/Article/1011148/soldier-missing-from-korean-waraccounted-for-hunt/> [retrived June 2017]