JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Self-Driving Car and Vehicle Data Acquisition System

Shubham Kumar Ojha, Siddharth Kumar, Mohit Singh,
Mr. Amitabh Srivastava, Mr. Mohd Shariq Mahboob
Department of Electronics and Communication Engineering
United Institute of Technology, Naini, Prayagraj, Uttar Pradesh (211010), India

Abstract: Self-driving cars have been developed by various companies such as Tesla, Audi, and Google, among others. The idea of autonomous cars originated from the increasing number of accidents caused by careless drivers. Our project aims to add a safety feature to vehicles, specifically a lane detection system, to prevent the car from swerving off the road when the driver is not in their senses. This feature can make driving easier and safer for everyone. We have gone through various articles and read papers on the topic to achieve this goal. Many companies and developers are showing interest in this field, and we are excited to be a part of it.

We used modern image processing techniques in combination with machine learning models to develop this feature. We utilized OpenCV, a Python library, to achieve our goals. We began with a simple lane detection module based on OpenCV to get our edges and lines in the image detected. However, the most significant challenge was interpreting these lines with the image to make sense of the lanes. To accomplish this, we used slope-intercept methods**.[2]**

Keywords: *Raspberry Pi 3B+ Board, RaspiCam, Arduino Uno.*

INTRODUCTION:

Self-driving cars are equipped with sensors that allow them to analyze their surroundings and make decisions without human interaction. These sensors enable the car to detect the pathway and road signals in its environment. Compared to traditional cars, the self-driving cars, market, which is highly cost-conscious. We have opted to use cameras rather than expensive offer several advantages such as reduced fuel wastage, increased safety, improved mobility, and customer satisfaction. One of the most significant benefits of self-driving cars is the reduction in traffic Distracted driving, impaired driving, and poor decision-making. By utilizing self-driving cars, which can make decisions and communicate with each other, it is expected that the number of accidents will decrease.

making. By utilizing self-driving cars, which can make decisions and communicate with each other, it is expected that the number of accidents will decrease.

Problem Statements: This project proposes the development of a low-cost prototype of a self-driving car equipped with a camera to detect traffic signals, such as turn right, turn left, and stop. Self-driving cars have the potential to significantly reduce the number of road accidents caused by human mistakes, which according to the

Association for Safe International Road Travel (ASIRT), result in an average of 3,700 fatalities and 20-50 million non-fatal injuries worldwide every day. This advanced technological development in the automotive industry not only provides drivers with comfort but also enhances safety on the roads Illustrates the annual number of road accident fatalities in different countries, with the light green color representing accidents most likely due to human mistakes. Self-driving cars are a promising solution to prevent such errors and have become increasingly popular in the market.[7]

Problem Solution: Our aim is to bring affordable self-driving technology to the Indian sensors to reduce costs and minimize the impact of customs. By leveraging programming techniques, we aim to develop a comprehensive business idea that provides services by installing our device in vehicles.

Our motivation for this project stems from the increasing demand for self-driving cars, especially given the electronic revolution in the automotive industry. Our prototype focuses on satisfaction. One of the most significant benefits of self-driving cars is the reduction in traffic accidents. Studies show that over 90% of all accidents are caused by human error, including

level 2 automation, which improves on the current industry standard by offering features such as over-the-network controls and object detection using cameras as opposed to costly lidar sensors. Our objectives for this project are to make daily driving safer, prevent car theft, save time with features such as auto-parking, and convert normal vehicles to level 2 automated cars. We intend to achieve this by introducing a kit that can be installed in any car, democratizing access to technology that is usually only available in high-end vehicles.

We recognize that achieving our objectives requires a high level of technical expertise, and we are committed to leveraging programming techniques and innovative solutions to make autonomous technology accessible to as many people as possible.

LITERATURE REVIEW:

Self-driving cars are a revolutionary advancement in the automotive industry, providing comfort and safety features for drivers. These cars rely on software-based systems that allow them to operate independently, using modules such as lane detection and object detection. The primary purpose of these modules is to prevent accidents caused by human error, which is responsible for the majority of non-fatal injuries that result in long-term disabilities The concept of self-driving cars dates back to the 1950s when General Motors proposed the "Firebird II," a self-driving model that used an electronic highway guidance system with radio-controlled electromagnetic fields. In 1958, John McCarthy suggested using artificial intelligence to develop self-driving cars rather than relying on lane detection. Today, self-driving cars rely on image processing techniques that involve feature extraction to interpret pixels and manipulate frames according to the vehicle's needs. One approach to lane detection is to detect lanes by thresholding the given image and calculating the direction based on white pixels. However, this technique is only applicable in a controlled environment where a path is created with white papers or similar material. Various improvements can be made to this method by implementing it.

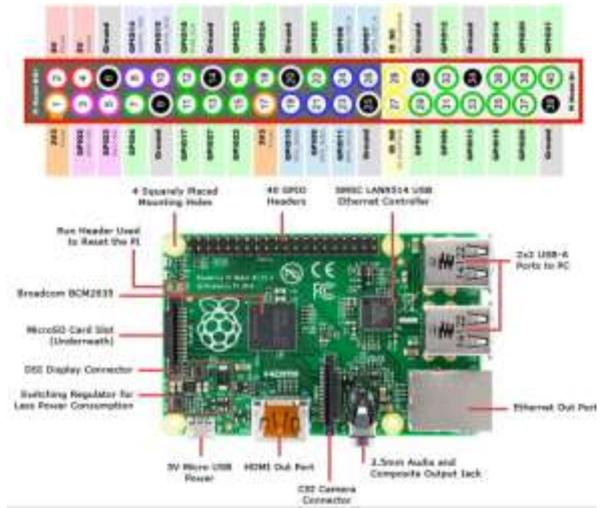
up, and interfaces with a camera to send frames to the server. Efficient image transmission is ensured by the pre-installed Python in the OS, while VNC Viewer is used to sending Python scripts and detection models additional steps or by using it in combination with other methods.

Object detection systems are designed to identify objects present in digital images or videos belonging to various classes, such as humans, cars, etc. The system requires several components, such as a model database, a feature detector, a hypothesis, and a hypothesis verifier, to perform object detection. However, without at least partial detection, segmentation cannot be applied, and therefore, the OD problem cannot be solved.

In conclusion, self-driving cars are an impressive technological development in the automotive industry. The modules they use, such as lane detection and object detection, provide comfort and safety features for drivers and passengers. Despite the challenges that still need to be addressed, self-driving cars have the potential to transform the way we travel and significantly reduce the number of accidents caused by human error**.[9]**

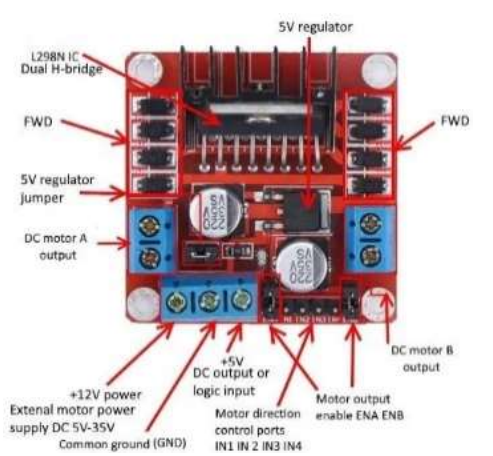
SYSTEM ANALYSIS AND DESIGN:

Raspberry Pi 3B+: The Raspberry Pi Foundation developed the Raspberry Pi, a range of small single-board computers, to promote computer science education in schools and developing countries. Despite its initial intended use in classrooms, the device's popularity has expanded to various fields, including robotics and weather monitoring, due to its low cost and portability. The device is delivered without peripherals, such as keyboards, mice, and a case. It includes a 1.4 GHz quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and power-over-ethernet compatibility. The Raspberry Pi is a reasonably priced Linux computer that offers a few general-purpose input and output (GPIO) pins that may operate electronic components for hands-on learning and experimentation. The Internet of Things. In this project, Python scripts control a car through a motor driver controller using the Raspberry Pi**.[4]**



Arduino Uno: The Arduino Uno is a powerful and versatile microcontroller board that is easy to use and affordable. It is ideal for hobbyists, educators, and professionals who want to develop interactive electronics projects. With its flexibility, compatibility with shields and low cost, the Arduino Uno has become one of the most popular boards in the Arduino family

L298N Motor Driver: The L298N Motor Driver is a well-known motor driver IC that is widely utilized in home projects and robotics. This versatile IC is designed to operate two DC motors or one bipolar stepper motor, and it can give up to 2 amps of current per channel. The integrated circuit is divided into three sections: power supply pins, control pins, and motor output pins. The power supply pins deliver power to the IC and motors, while the control pins control the motors' direction and speed. The motor output pins are connected to the motors directly. The Top of the FormL298N Motor Driver can handle large currents and voltages and includes built-in protective measures to protect the IC and the motor.**[1]**



Camera (RaspiCam 2 with CSC Cable): The RaspiCam 2 camera module is a top-quality device that is specially designed for use with Raspberry Pi single-board computers. It is an upgraded version of its predecessor, the RaspiCam, and it comes packed with an array of advanced features that make it suitable for a wide range of applications, including computer vision projects, robotics, and surveillance systems. With its 8-megapixel sensor and 62.2-degree field of view, the camera captures images in vivid colors and stunning detail. Additionally, the camera module offers automatic white balance, automatic exposure control, and digital zoom for improved imaging capabilities. It also features different modes of operation that can be switched easily using Raspberry Pi software, making it highly versatile and user-friendly.

PCB DESIGN:

This chapter focuses primarily on the design of a printed circuit board (PCB) for a traffic light system. The PCB design involves creating a circuit layout that incorporates various electronic components such as LEDs, resistors, capacitors, and transistors. These components work together to enable the traffic light system to function efficiently. The design process entails using specialized software tools to create a schematic diagram and a physical layout of the PCB. It also involves following specific design rules and guidelines to ensure the PCB meets industry standards and is reliable. Overall, this chapter provides an overview of the key considerations involved in designing a traffic light PCB.**[1]**

Component:

1. Zero PCB
2. CD4017 IC
3. IN4007 DIODE (8)
4. 56-ohm Resistor
5. 220-ohm Resistor
6. 100k ohm Resistor
7. NE555 IC
8. 100uf Capacitor
9. 18k ohm Resistor
10. 9V battery

How to Make Traffic Light Signals:

First, connect the NE555 IC in stable mode to generate a clock signal for the CD4017 IC. Use the 18kΩ resistor, 100uf capacitor, and 220Ω resistor to set the frequency of the clock signal to around 1Hz.

Connect the output pins of the NE555 IC to the clock input pin (pin 14) of the CD4017 IC.

Connect the 8 output pins of the CD4017 IC to the anodes of the 8 LEDs (3 for each color and 1 for the Zero PCB)

Connect the cathodes of the LEDs to the 220Ω resistor and then to the ground.

Connect the 1N4007 diodes in parallel with each of the 220Ω resistors and the LEDs. This will prevent reverse voltage from damaging the LEDs.

Connect the 56Ω resistor in series with the anode of the Zero PCD LED. This will limit the current flowing through the LED. Connect the positive terminal of the 9V battery to the power supply rail and the negative terminal to the ground. Connect the positive terminal of the 9V battery to the power supply rail and the negative terminal to the ground.

Once all the components are connected as described above, the circuit will function as a traffic light. The LEDs will light up one at a time, simulating a traffic light sequence. He sequences will repeat continuously. Here is the sequence of the lights:

Zero PCB LED is ON.

Red LED is ON.

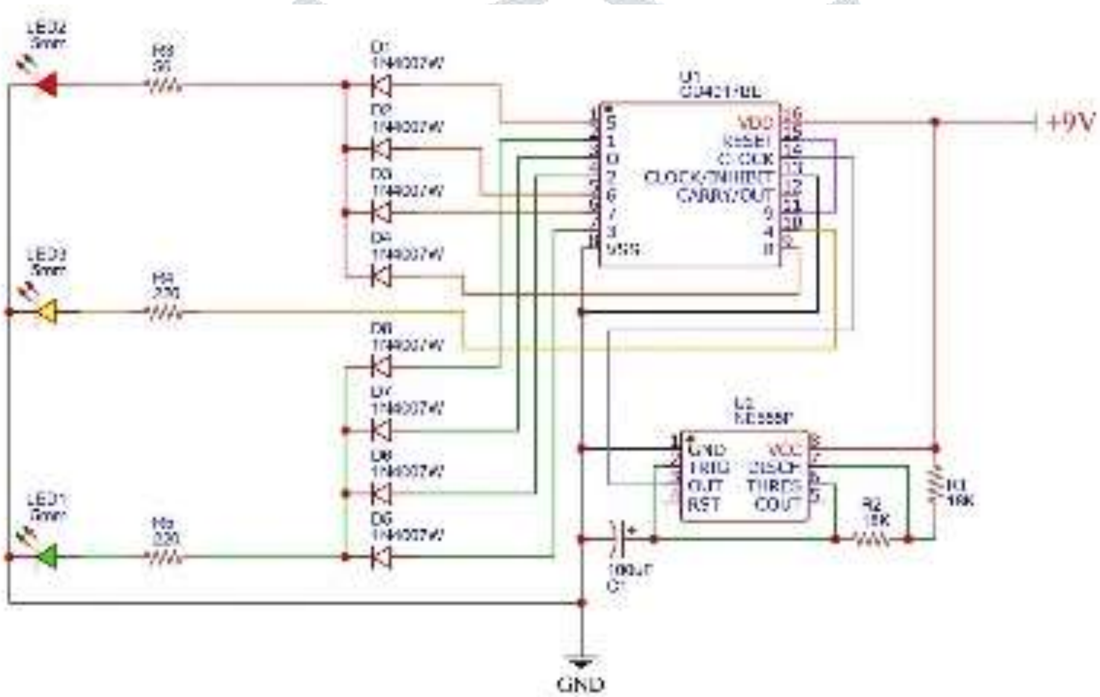
Red and Yellow LEDs are ON.

Green LED is ON.

Yellow LED is ON.

Red LED is ON.

The duration of each LED being ON can be adjusted by changing the values of the resistors and capacitors in the circuit.



SOFTWARE & LIBRARIES

The project requirements explicitly state that all software and libraries that will be utilized should be mentioned in a clear and concise manner. It is essential to ensure that all relevant information regarding the software and libraries is included to avoid any confusion and misunderstandings throughout the development process. The explicit mention of the software and libraries to be used will also allow for proper documentation, maintainability, and ease of troubleshooting any issues that may arise. Therefore, it is crucial to adhere to the project's guidelines and ensure that all software and libraries are stated explicitly.**[1]**

Software:

1. Raspbian OS.
2. VNC Viewer
3. Putty
4. Arduino IDE

Libraries:

1. OpenCV
2. Pi Camera
3. IO- Python

INSTALLATION PROCESS TO SOFTWARE AND LIBRARIES

This chapter provides a detailed and concise explanation of the installation process for all the required software and libraries. The explanation is designed to be clear and easy to follow, with step-by-step instructions on how to complete the installation process successfully. It is crucial to ensure that the installation process is correctly carried out, as any errors or misconfigurations could result in significant issues further down the line.

Therefore, the chapter's clear and comprehensive instructions are vital in guaranteeing that all required software and libraries are installed correctly and efficiently, ensuring that the project can progress without any hitches.**[1]**

Software:

1. Raspbian OS
2. Arduino IDE
3. Putty

Libraries:

1. Open CV
2. Pi-Camera

METHODOLOGY

System Architecture: Self-driving cars utilize a combination of sensors and cameras to navigate roads and detect their surroundings. The visual information provided by the cameras is processed using image processing techniques, which involve extracting features from each frame of the video captured by the camera. To process the video data, each frame is first converted into a black-and-white threshold image using the thresholding technique. The edges are then detected using the Hough lines method, which collects collinear points into an array of arrays that are used as drawing points to draw straight lines. The Hough lining technique allows for greater flexibility in detecting different types of lines by setting the orientation and length of the lines to be detected. Once the lines are drawn, the angle of the lines is calculated, and a decision is made on whether to turn left or right based on the orientation of the lines. In summary, self-driving cars heavily rely on image processing techniques to perceive their environment, make decisions, and navigate roads safely.**[5]**

Design Constraints: Our autonomous vehicle prototype is equipped with advanced image processing technology that enables it to detect lanes and obstacles while also featuring smart control and smart lock capabilities. The design of this project was intended to perform multiple tasks using just a single camera, making it a cost-effective and efficient solution in the market. Although some sensors were excluded to reduce costs, we chose to use an Arduino UNO over the Raspberry Pi's GPIO pins to control the servo motors, which require 12V DC, while the Raspberry Pi only operates on 5V. While the camera used in the prototype is a cheaper 5 MP option, more expensive alternatives with night vision and higher resolutions are also available. The installation of OpenCV Python on the Raspberry Pi was a significant challenge during the development process, as was the voltage difference between the Raspberry Pi and electronic components, which was resolved by utilizing an Arduino to power the parts.

Design Methodology: The project's core libraries include CV2, NumPy, and math, which serve as the foundation of the autonomous vehicle prototype. CV2 is a popular open-source computer vision library used for image processing and analysis, and we have used it to detect obstacles and lanes on the vehicle's path. NumPy is a powerful library that provides support for multi-dimensional arrays and mathematical functions used to manipulate image arrays generated by CV2. Math is another Python built-in library we have used for various mathematical operations to calculate distances and angles for navigation. The time library is also utilized to introduce delays between tasks. Overall, these libraries' collective capabilities have enhanced the project's functionality and performance, reducing development costs and time.**[4]**

Result Working Module:

[6]

Figure.

The above figure shows the obstacle detection

[3]

Figure

The above figure shows the lane-end detection

[1]

Figure

The above figure shows the traffic light detection

CONCLUSION:

Self-driving cars and vehicle data are two of the most significant technological advancements of our time. The emergence of autonomous vehicles has the potential to revolutionize the way we commute, making our roads safer and more efficient. At the same time, vehicle data is playing an increasingly important role in optimizing vehicle performance, reducing costs, and improving safety. In this article, we will explore these two topics and summarize their basics.

Self-Driving Cars:

Self-driving cars, also referred to as autonomous vehicles, have the ability to navigate roads without human intervention, using a range of technologies such as sensors and cameras to gather data and make instantaneous decisions. These vehicles have the potential to significantly reduce traffic accidents, as human error is the primary cause of such incidents. Moreover, self-driving cars have the ability to optimize their routes using advanced algorithms, which can reduce congestion and travel times. Ultimately, this technology has the power to revolutionize the way we approach transportation, with the potential for on-demand car summoning and a reduction in the need for parking spaces, thus freeing up space for alternative urban planning projects.

Vehicle Data:

As the world becomes more connected and automated, the importance of vehicle data is growing rapidly. This data encompasses a wide range of information, including engine performance, fuel consumption, driver behavior, and vehicle location. Fleet managers can use this data to optimize performance and reduce costs, while companies can identify training opportunities to improve driver safety and reduce accidents. Additionally, transportation authorities can use vehicle data to enhance road safety by identifying problem areas and implementing measures to reduce the risk of accidents. Moreover, ride-hailing companies and delivery companies are leveraging vehicle data to support their business models by matching riders with drivers, optimizing routes, and tracking the location of their vehicles.

Limitation:

1. Night vision
2. Cyber Attack
3. Driver Supervision
4. Weather Conditions

Future Works:

1. Curved Lane Detection
2. Over Network Control
3. Security features
4. Including Deep learning
5. Real car Implementation
6. Object Detection

REFERENCES:

1. *Zhenchao Ouyang, Jianwei Niu, Yu Liu and Mohsen Guizani, "Deep CNN-based real-time traffic light detector for self-driving vehicles", IEEE transactions on Mobile Computing, vol. 19, no. 2, pp. 300-313, 2019.*

2. *Mohamed AA Babiker, Mohamed AO Elawad and Azza HM Ahmed, "Convolutional neural network for a self-driving car in a virtual environment", 2019 International Conference on Computer Control Electrical and Electronics Engineering (ICCCEEE), pp. 1-6, 2019.*

3. *Brilian Tafjira Nugraha and Shun-Feng Su, "Towards self-driving car using convolutional neural network and road lane detector", 2017 2nd international conference on automation cognitive science optics micro electro-mechanical system and information technology (ICACOMIT), pp. 65-69, 2017*

4. *Syed OwaisAli Chishti, Sana Riaz, Muhammad BilalZaib and Mohammad Nauman, "Self-driving cars using CNN and Q-learning", 2018 IEEE 21st International Multi-Topic Conference (INMIC), pp. 1-7, 2018.*

5. *P. G. Chaitra, V. Deepthi, S. Gautami, H. M. Suraj and Naveen Kumar, "Convolutional Neural Network based Working Model of Self Driving Car-a Study", 2021*

6. *International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 645-650, 2020.*

7. *Aman Bhalla, Munipalle Sai Nikhila and Pradeep Singh, "Simulation of self-driving car using deep learning", 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), pp. 519-525, 2020.*

8. *Hesham Alghodhaifi and Sridhar Lakshmanan, "Autonomous vehicle evaluation: a comprehensive survey on modeling and simulation approaches", IEEE Access, 2021*