Electronics and communication enginering

Self Controlling Smart Vehicle

Theme based project

Abstract

With the highly rising traffic congestion , human drivers errors due to several reasons, we have seen an increasing number of accidents and inconvenience on the road. With hardcoded rules autonomous cars are very unlikely to make blunders like human drivers. Unlike human drivers, autonomous cars will be able to operate with the same efficiency and decision making under any situation. Hence, to reduce traffic congestion, front collision, provide relaxed travelling experience, provide increased safety and better observe traffic rules, autonomous cars prove to be an ideal solution The report proposes a self-driving car model also called autonomous, robotic or driver-less car is one that operates and navigates using its intelligence. The basic idea is to develop a car to portray an automated car. The model consists of the following software and hardware components such as CNN (Convolutional neural network), MIT app inventor, Arduino, and an Ultrasonic sensor. The (CNN) convolutional neural network is used to detect 41 different traffic signs and we introduced diiferent modes through MIT app

1 Introduction

Estimates suggest nearly 80 percent of car crashes are due to human error. Self-driving cars can eliminate the possibility of human error and hence reduces the number of accidents In the year 2019, Mr Nitin Gadkari, the Road and Transport Minister of India said that he won't allow driverless cars in India as it will result in job losses. If driverless cars are allowed in India, they can be used for a variety of other purposes also such as delivery or pickup service, which will also save time and human resources. The implementation of autonomous cars in India can address several critical needs and challenges in the country's transportation system. Firstly, safety is a significant concern on Indian roads, where accidents and fatalities are unfortunately common. Autonomous cars, equipped with advanced sensors and artificial intelligence, have the potential to minimize human errors and improve overall road safety. With their ability to adhere strictly to traffic rules, maintain safe distances, and quickly respond to potential hazards, autonomous cars can greatly reduce accidents and save lives.

Moreover, India's cities often suffer from severe traffic congestion, resulting in increased travel times, fuel consumption, and pollution. Autonomous cars can play a crucial role in optimizing traffic management by leveraging real-time data, predictive analytics, and intelligent routing algorithms. Through communication with other autonomous vehicles and traffic infrastructure, these cars can actively avoid congestion, select the most efficient routes, and coordinate their movements. As a result, they can significantly reduce travel times, alleviate congestion, and contribute to smoother traffic flow.

the implementation of autonomous cars in India has the potential to address critical needs such as safety, traffic congestion, accessibility, and sustainability. By leveraging advanced technologies and intelligent systems, autonomous vehicles can revolutionize the transportation landscape, making it safer, more efficient, and accessible for all.
.

1.1 Literature Survey

From [a] we studied basic approach methods and overview of cnn algorith for traffic detection
from [b] we studied about pi 4 required commands and process to operate it.
. [c] we found autonomouse vechile in hardware implementation view

1.2 Problem Statement

we have seen an increasing number of accidents and inconvenience on the road. With hardcoded rules autonomous cars are very unlikely to make blunders like human drivers. Unlike human drivers, autonomous cars will be able to operate with the same efficiency and decision making under any situation. Hence, to reduce traffic congestion, front collision, provide relaxed travelling experience, provide increased safety and better observe traffic rules, autonomous cars prove to be an ideal solution.

1.3 Outcomes

1. obstacle avoidance
2. path follower
3. voice commands
4. manual mode
5. traffic sign detection
6. Speed control

---

2 Making Process

2.1 Apparatus

2.1.1 Hardware

2.1.2 Peripherals

1. DC Gear Motor x 4,
2. Arduino UNO,
3. IR Sensor x 2,
4. L298 Motor Driver,
5. HC-05 Bluetooth Module,
6. sg90 servo motor,
7. Chassis Board wooden x 2
8. Ultrasonic Sensor Holder,
9. Ultrasonic Sensor hc-sr04,
10. 4 Robot Car Tyres Wheels,
11. Male to Female jumper Wires,
12. On/Off Switch,
13. Battery Holder – 2 Cell ,
14. Battery Cell 3.7V x 2
15. pi cam

2.1.3 Microcontrollers and processor

16. Aurdino uno
17. Raspberry pi 4

2.1.4 Software

18. Arduino IDE
19. Raspbian OS
20. MIT app Inventor

2.2 Flow of solution

2.2.1 Vehicle and hardware part

The autonomous car, built using Arduino, an ultrasonic sensor, servo motor, and IR sensors, can be integrated with the MIT App Inventor to enhance its functionality. By creating a mobile app interface using the MIT App Inventor, users can remotely control and monitor the car. The app allows for designing a user-friendly interface with buttons, sliders, or joysticks to control the car's movements. Through a wireless connection established between the app and the Arduino board, commands are sent from the app to the car. The Arduino code is modified to interpret these commands and actuate the necessary actions, such as steering or obstacle detection. This integration provides an intuitive and convenient way to interact with the autonomous car, allowing users to control its movements and access sensor data easily. The MIT App Inventor complements the hardware setup, enhancing the overall functionality and user experience of the autonomous car project.

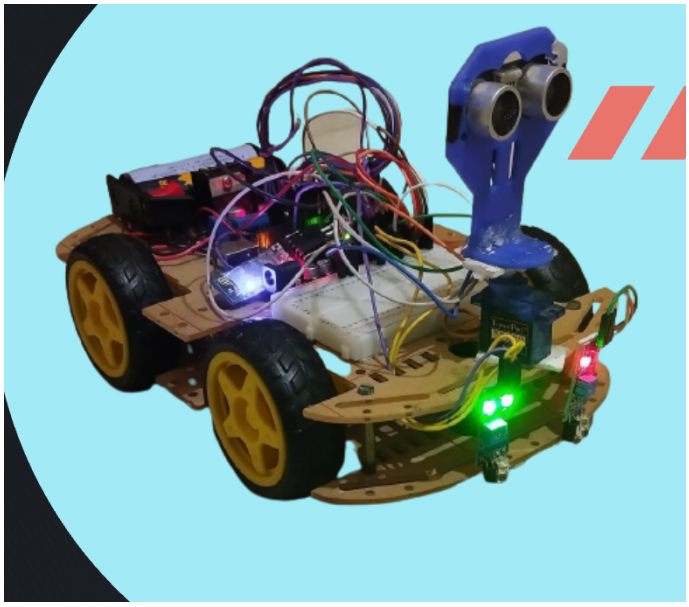


Figure 1: Vehicle setup

The autonomous car built using Arduino, an ultrasonic sensor, servo motor, and IR sensors is a self-driving vehicle designed to navigate its environment independently. The ultrasonic sensor allows the car to detect obstacles in its path, while the servo

motor controls the steering mechanism, enabling the car to change direction accordingly. The IR sensors play a crucial role in line following or edge detection, enabling the car to stay on track. With the Arduino board as its brain, the car processes sensor inputs and makes decisions based on programmed algorithms. It offers an exciting platform for experimentation and learning in the field of robotics and autonomous systems.

Integrating the autonomous car with the MIT App Inventor can enhance its functionality by allowing remote control and monitoring through a mobile app. The MIT App Inventor is a visual programming environment that simplifies the creation of Android apps. Here's an overview of how you can integrate the autonomous car with the MIT App Inventor:

Design the user interface:

Open the MIT App Inventor and create a new project. Design the user interface elements, such as buttons, sliders, or joysticks, based on the desired control features for the autonomous car. Establish a connection between the app and the autonomous car:

Determine the communication protocol between the app and the Arduino board (e.g., Bluetooth, Wi-Fi). Select the appropriate module or shield for the Arduino board to establish the chosen wireless connection. Write the Arduino code for communication: Adapt the existing Arduino code to listen for commands received from the MIT App Inventor. Define the necessary functions or routines to interpret the commands and control the autonomous car accordingly. Program the app in MIT App Inventor:

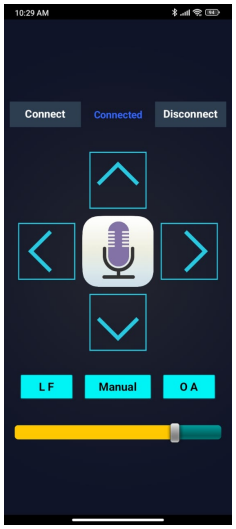


Figure 2: MIT app interface

Use the blocks-based programming interface of MIT App Inventor to define the app's behavior. Set up the communication protocol and commands to send to the Arduino board. Map the app's user interface elements to the corresponding commands. Test and refine:
Install the app on an Android device. Establish the wireless connection between the app and the Arduino board. Test the control functionality by sending commands from the app and observing the autonomous car's response. Fine-tune the app and Arduino code as needed for optimal performance. By integrating the autonomous car with the MIT App Inventor, you can control the car's movements, monitor sensor data, and even implement additional features like real-time video streaming or data logging. The MIT App Inventor provides a user-friendly platform to create a customized mobile app interface that enhances the overall functionality and interaction with the autonomous car.

2.3 Traffic sign detection

Utilizing a CNN (Convolutional Neural Network) for a traffic sign detection project offers an efficient solution for automatically recognizing and classifying traffic signs in images. By collecting a labeled dataset of diverse traffic sign images and preprocessing them through resizing and normalization, the stage is set for training the CNN model. Selecting a suitable architecture and adjusting it to match the input size and number of output classes, the model is trained using the dataset, optimizing its performance through hyperparameter tuning and data augmentation. Evaluation on a validation set ensures the model's effectiveness, and subsequent testing on a separate dataset validates its real-world performance. Once deployed in the desired environment, the CNN-based traffic sign detection system can be fine-tuned and improved over time, ensuring compliance with safety regulations and standards.

CNN's have been gaining popularity in the past couple of years due to their ability to generalize and classify the data with high accuracy. , we trained traffic signs with over 30000 images of 43 different classes with the help of TensorFlow and Keras.

2.3.1 Data set

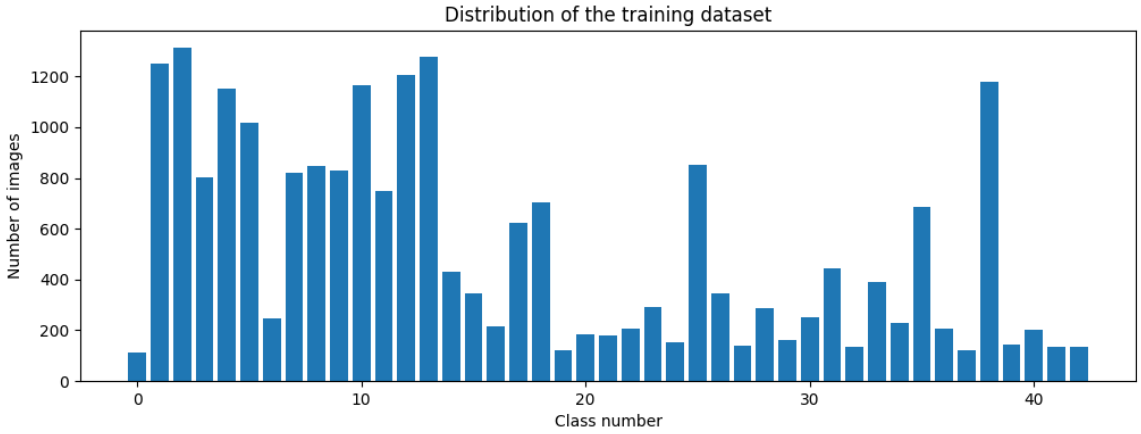


Figure 3: Data set distribution

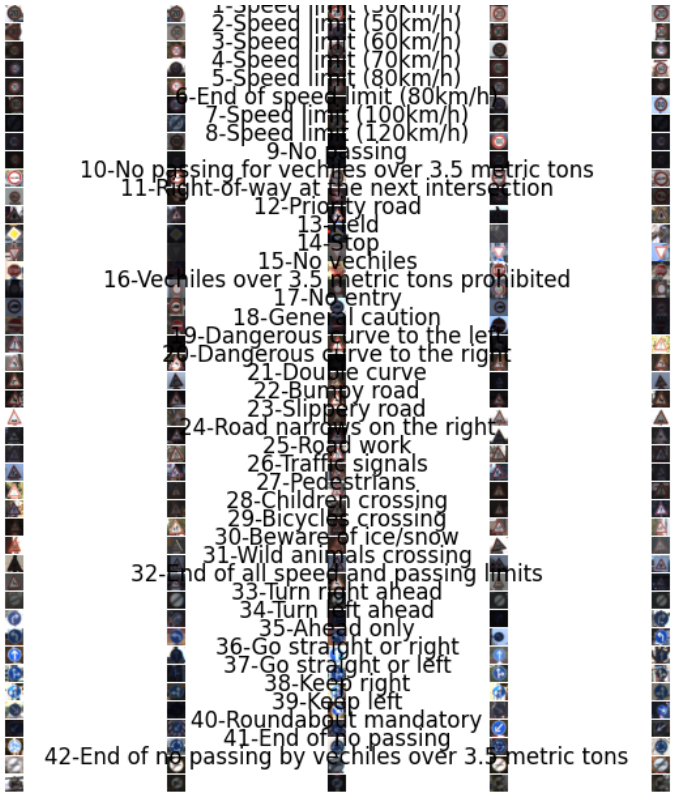
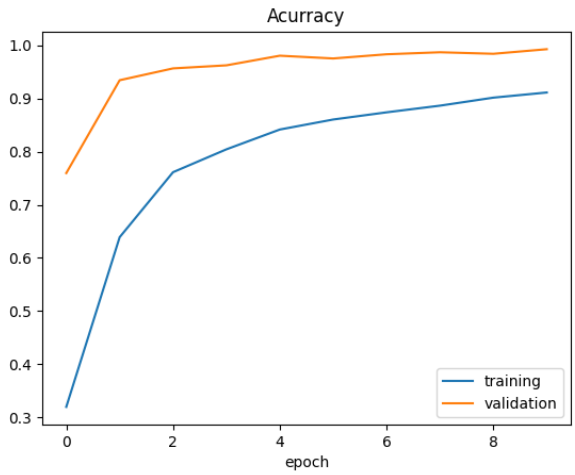
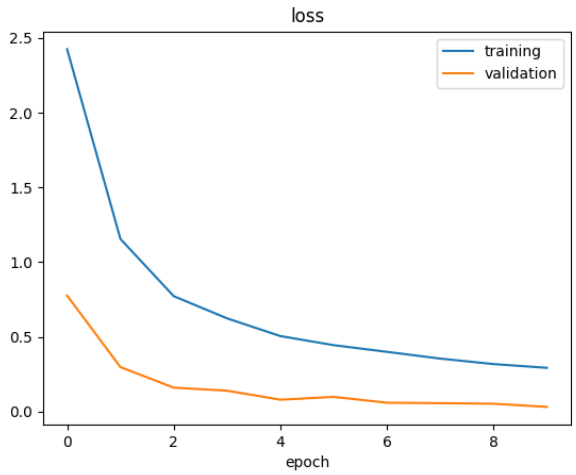


Figure 4: Data set labels

Performance Graphs :





...

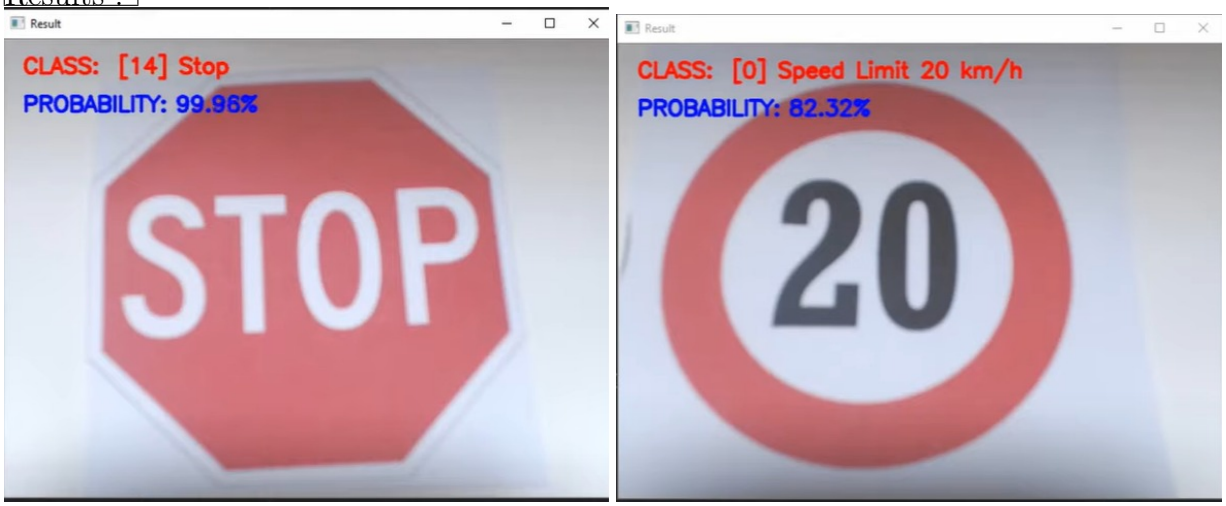
In graphs plotting the accuracy and loss during the training of a CNN (Convolutional Neural Network), the epochs are used as the x-axis. The epoch value represents the number of times the entire training dataset has been iterated through during the training process.

The use of epochs in these graphs is to visualize how the accuracy and loss metrics change over time as the model undergoes training. Each epoch corresponds to a point on the graph, showing the model's performance at that particular stage. By observing the trends and patterns in the graphs, one can gain insights into how the model is learning and improving over successive epochs.

These graphs allow for monitoring the convergence of the model, identifying underfitting or overfitting, and determining the optimal number of epochs for training. They help in assessing the progress of the model's learning and making informed decisions about further training or adjustments to achieve the desired accuracy and minimize loss.

2.4 Results :

Results :



...

3 CONCLUSION:

Automation in cars in real world is a very big field, where many sensors come into action, our idea is to solve many of those into two sensors by detecting distance and objects like stop sign, signals and other obstacles with a single method of monocular vision which can be enhanced in future from a scaled car to an actual car. The prototype focuses on these functionalities which we are developing in a model rc car while others focus on only one aspect of it.

4 REFERENCES:

1. https://www.computervision.zone/cours...
2. https://www.raspberrypi.com/documentation/
3. https://ieeexplore.ieee.org/document/8220479 .
4. Design and Implementation of Autonomous Car using Raspberry Pi International Journal of Computer Applications (0975 – 8887) Volume 113 – No. 9, March 2015

5. Syed Owais Ali Chishti, Sana Riaz, Muhammad Bilal Zaib, Mohammad Nauman, “Self-driving Cars Using CNN and Qlearning”, in 2018 IEEE 21st International Multi-Topic Conference (INMIC)

6. M. A. Hossain, M. S. Rahman, M. M. Hassan, M. S. Islam, and K. Ahmed, ”Autonomous Car Navigation Using Deep Learning Techniques,” 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 2020, pp. 1537-1542.

5 TEAM MEMBERS:

1. 1602-20-735-177varshasri
2. 1602-20-735-082 CH.Manoj reddy