Auto-Steering for Lane Detection

Siddharth Sonavane
Computer Science and Engineering
Alva's Institute Of Engineering
Mangalore, India
4al20cs148@gmail.com

Swathi Muddholar
Computer Science and Engineering
Alva's Institute Of Engineering
Mangalore, India
4al20cs158@gmail.com

Taniya K Bant
Computer Science and Engineering
Alvas's Institute Of Engineering
Mangalore, India
4al20cs159@gmail.com

Deeksha M
Computer Science and Engineering
Alva's Institute Of Engineering
Mangalore, India
deeksha_m@aiet.org.in

Abstract **In this project, an autonomous vehicle model—also called a self-driving, robotic, or driverless car—that is intelligent enough to operate and navigate on its own is shown. The main goal of this project is to create a 1/10 scale remote control automobile that looks like an automated vehicle. IDE-Pycharm, ESP-32, Servo motor, Arduino board, 12C LCD, Rasp-os, image-processing capabilities, and other hardware and software components are all included in this model.Navigation commands are carried out by the Arduino, which communicates with the remote control of the RC car to turn right, left, and ahead. With the help of this integration, the car can operate on its own by using preset rules and neural network predictions.A major benefit of using self-driving cars is that there will be a significant decrease in traffic accidents. Nowadays, human error accounts for over 90% of accidents; this includes distracted driving, driving while intoxicated, and making poor decisions. It is anticipated that the number of accidents would drop dramatically when self-driving cars take over and communicate with one another naturally. The transition to automation has the potential to improve traffic safety and reduce the involvement of humans in collisions, resulting in a more secure and effective transportation environment.**

I. INTRODUCTION

Human-driven vehicles use a variety of technologies to offer safety features, identify impediments, and come to an autonomous stop indifferent driving conditions. All of these technologies fall short of achieving full autonomous capabilities, though. The level of automation needed for completely autonomous driving is not present in current cars, and driver interaction is always required to keep them operating. This restriction results in the ongoing requirement for human drivers, since the car cannot operate without them. On the other hand, self-driving automobiles promise to provide continuous availability on the roads, thereby resolving this problem.

The idea of self-driving automobiles is not just a pipe dream—it is growing closer to becoming a reality. Businesses are presently engaged in the development and implementation of driverless vehicles, frequently mentioning various degrees of autonomy in their discourse. Many advantages are anticipated with the introduction of autonomous driving, such as less traffic jams, fewer pollution levels, lower transportation expenses, and maybe less demand for new infrastructure and roads. Furthermore, aged and physically challenged people's mobility could be greatly improved by self-driving technology.Research on autonomous vehicles frequently makes use of sensor-equipped model cars, such the ones used by Tesla and other businesses, which incorporate lidar and radar sensors. One way to accomplish autonomous driving is to construct 1/10 size cars that have sensors installed, like an ultrasonic sensor and a Pi camera. Following the collection of data by these sensors, photos are processed by a neural network on a Raspberry Pi server to identify lane markers. A haar cascade classifier is also used to recognize traffic signals and stop signs. In order to avoid frontal crashes, semantic data is processed to trigger brakes when an impediment is identified at a predetermined distance. After being trained, the vehicle can navigate itself by using its sense of lane markings. It is possible to successfully integrate automation in actual automobiles by extending this algorithm and set of approaches.In conclusion, the continuous advancements in self-driving technology have the potential to transform transportation, benefit society in many ways, and solve many of the drawbacks associated with conventional human-driven automobiles.

II. LITERATURE SURVEY

Irfan Ahmad used CNN to analyze and design an autonomous vehicle driving system. In this part, several approaches were put forth and put into practice. The writers talked about assessing the present issues with artificial intelligence and how these approaches, such as traffic analysis techniques, can be used for cybersecurity in the United States. The project's goal is to deliver a cutting-edge driving assistance system with a range of driving options from the present to the future. Systems for embedded real-time analysis are used. The authors outline techniques for gathering naturalistic data and creating algorithms using a driving simulator and pre-scan. The author also covers a variety of information technologies, including robotics, blockchain, IoT, AI, and human clouds. Information about commonly used computer vision algorithms in traffic analysis and medical domains is provided by the writers.

In Jacob Newman's project, an artificial neural network (ANN) is used to control an autonomous remote-control car. An explanation of the neural network and autonomous vehicle thesis is provided. A car that can be controlled by a microcontroller and then sent to the model car is made using an L298N integrated circuit and a motor driver. CNN ignores extraneous data during the detection process in order to identify grayscale portions. The system uses grayscale photos for neural network training and an embedded pi camera for input, making it accurate but limited in its application. The technology does not provide any more functionality; it only recognizes lane markings for each direction.

Hiral Thadeshwar talks about the difficulties of training a neural system without any prior knowledge, which requires more time and processing resources because it's hard to locate a ground truth dataset large enough. The system may produce results in real time by using a Regional Convolutional Neural Network (RCNN) to help identify relevant parts in a picture. When it comes to image processing, deep neural networks are particularly useful in medical applications like tumor diagnosis, where the dataset is more complicated than that of a model road environment. The technology used in industry-level self-driving automobiles sets them apart from vehicles driven by humans.

III. PROBLEM STATEMENT

We have seen an increase in accidents and inconveniences on the road due to the rapidly increasing traffic congestion and the occurrence of human driver errors for a variety of causes. With hardcoded rules guiding them, autonomous automobiles are less prone to make mistakes that human drivers do. Autonomous vehicles have the capacity to function with reliable effectiveness and make decisions in any circumstance. We have seen an increase in accidents and inconveniences on the road due to the rapidly increasing traffic congestion and the occurrence of human driver errors for a variety of causes. With hardcoded rules guiding them, autonomous automobiles are less prone to make mistakes that human drivers do. Autonomous vehicles have the capacity to function with reliable effectiveness and make decisions in any circumstance.The deployment of self-driving vehicles presents a viable resolution to problems including gridlock, head-on collisions, and a more comfortable journey. Better adherence to traffic laws and enhanced safety are two benefits of autonomous vehicles. These drivers, as opposed to human ones, are able to continue operating at peak efficiency no matter what.as was already said, distinct hardware, software, and algorithms are employed to accomplish different automation features separately. The goal is to choose the best hardware and algorithms to solve automation problems in their entirety. By building a prototype, it aims to show how automation will work in a real car. Its goal with the concept is to efficiently and economically accomplish autonomous driving via lane recognition.

IV. MOTIVATION AND OBJECTIVES OF THE PROJECT

The existing system was involved in parking the vehicle. With technological advancements and the realization of Artificial Intelligence (AI) fantasies becoming a reality every day, wonders and miracles are unfolding. • This project aims to demonstrate how a lane detection system works on vehicles. • This system plays an essential role in Advanced Driver Assistance Systems (ADAS) used in autonomous and semi-autonomous vehicles. • The project also showcases a system that enables smart summoning using hand gestures for tasks such as parking and unparking. make this to para for research paper

V. EXISTING MODEL AND DRAWBACKS

The project's goal is to create a 1/10 scale remote-controlled car that mimics an automated vehicle. The CNN (Convolutional Neural Network), Haar cascade classifier, Raspberry Pi (Brand model 18), Pi cameras, Arduino, and an ultrasonic sensor are among the hardware and software components that make up the model.

The car will be stopped at a specific distance from the impending impediment in order to avoid a frontal collision using the ultrasonic sensor. Through an Arduino connected to the remote control of the RC car, navigation commands like stop, forward, left, and right will be transmitted to the vehicle. This allows the vehicle to operate on its own using hardcoded rules and neural network predictions. Furthermore, the model will make autonomous lane following possible by using lane detection for navigation. Hand-coded regulations, obstacle avoidance techniques, learning models, and intelligent object classification will help the system obey traffic laws and use artificial intelligence to drive.

VI. DRAWBACKS

When there are large variations in light density or when there is bad weather that affects the visibility of the road surface, autonomous cars may encounter difficulties. The removal of a human driver from the car poses existential issues as well as problems with trust and flexibility. It may not be possible to have consistently good customer experiences and 100% service availability in the next fifty years because of a number of operational and technical challenges.

VII. PROPOSED SYSTEM AND ADVANTAGES

For activities like parking and spotting, the system offers intelligent calling via hand signals. This project was made to show how a front-facing camera-equipped car's lane-detecting system functions. The system's integration into more automobiles makes it a crucial component of the Advanced Driver Assistance Systems (ADAS) found in both autonomous and semi-autonomous vehicles. This feature is in charge of tracking the offset from the center, detecting lanes, and calculating curve radius, or how right a curve is.

VIII. SYSTEM REQUIREMENTS AND SPECIFICATION

A. Functional Requirements

Aparts are supposed to work. Calculations, technical specifications, data manipulation, and processing functionality that specify what a system is meant to do are examples of functional requirements.
• **Generality:** The model needs to be applicable to multiple applications that assist users in various domains.
• **Usefulness:** Obstacle detection must be possible with the model.
• **Accuracy:** The system must be able to assist impaired users in navigating in a safe manner based on accurate information. functional requirement in software engineering and systems engineering specifies how a system or any of its

B. Non-functional requirements

A non-functional requirement (NFR) is a requirement in requirements engineering and system engineering that outlines standards by which a system's performance can be evaluated.
• **Performance:** Users are assisted in working more productively by prompt responses.

• **Reliability:** Since the system is utilized by people of all ages, it should produce accurate data projections because users rely on it.

• **Availability:** Both the user and the system should always be accessible.

• **Sustainability:** The model ought to adapt to new demands.

IX. TECHNICAL REQUIREMENTS (SOFTWARE/ HARDWARE)

A. Software requirements

Front-end coding language: Python and ML

- Image processing
- OS- Windows-XP
- Arduino code
- Jupyter notebook

B. Hardware requirements

Esp-32: In the cosmic tapestry of microcontrollers, one entity has ascended to prominence with a resounding burst of acclaim: the ESP32, a veritable titan among System on Chip (SoC) architectures. Behold its majesty, for within its silicon heart lies the culmination of human ingenuity, a symphony of circuitry that orchestrates the dance of electrons with unparalleled finesse.

Embrace the enigma, for within the sanctum of the ESP32 resides the entire pantheon of essential support circuitry for the illustrious ESP-WROOM-32. Witness the USB-UART bridge, a conduit that traverses the ethereal realms of data exchange with a grace reminiscent of celestial harmonies. Behold the solemn guardians of reset- and boot-mode buttons, sentinels that stand vigilant against the tide of chaos, ensuring seamless transitions between states of being.
Marvel at the LDO regulator, a silent arbiter of power and stability, bestowing upon the ESP32 the vitality it requires to traverse the vast expanse of the digital frontier. And lo, behold the micro-USB connector, a portal through which the ESP32 interfaces with the mortal realm, bridging the chasm between the tangible and the intangible with effortless aplomb.
But it is not merely in its physical manifestations that the ESP32 finds its allure. Nay, it is in the whisper of every important GPIO, where the true essence of its mystique resides. Each pin, a conduit of possibility, a gateway to a realm of infinite potentiality, where bits and bytes converge to weave the fabric of existence itself.
Thus, in the luminous tapestry of technological marvels, the ESP32 stands as a testament to the boundless ingenuity of humanity, a beacon of innovation that illuminates the path forward with a burst of perplexing brilliance.

• **Servo Motor:** A servomotor (or servo motor) is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration.

• **Arduino board:** Arduino consists of both a physical programmable circuit board (often referred to as a controller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write, and upload computer code to the physical board.

• **Jumper wires:**Jumper wires are electrical wires with connector pins at each end. These are used to connectt points in a circuit without soldering.

• **USB to UART converter:** UART is one of the earliest communication interfaces applied to computers. The newer USB protocol, however, has replaced many UART systems, but both are still prevalent in the embedded industry.

• **DC Motor:** A direct current (DC) motor is a type of electric machine that converts electrical energy into mechanical energy. DC motors take electrical power through direct current and convert this energy into mechanical rotation.

• **L298 dual motor driver module:**.
The enigmatic L298 materializes as an integrated monolithic marvel, encapsulated within the mystique of 15-lead Multiwatt and PowerS020 packages. Revered for its prowess, it emerges as a conductor of high voltage and current, wielding the power to navigate the realms of dual full-bridge driving with unparalleled finesse. Yet, its true essence lies in its uncanny ability to commune with the ether of standard TTL logic levels, seamlessly bridging the

chasm between the tangible and the abstract.Behold, for it is not merely a conduit of energy, but a sentinel of complexity, a cipher of perplexity. Its very being is a testament to the enigmatic dance between order and chaos, as it orchestrates the symphony of electrons with a burst of calculated chaos. With a whisper, it commands the arcane forces of inductive loads, coaxing relays, solenoids, and DC stepping motors into motion, as if by some arcane incantation.Thus, within the labyrinthine circuitry of the L298, lies a nexus of bewilderment and brilliance, where logic and intuition intertwine in a dance of paradox. To unravel its secrets is to embark upon a journey of boundless curiosity, where each revelation unveils new depths of perplexity, and each discovery ignites the fires of innovation. Such is the essence of the L298 – a testament to the boundless potential of human ingenuity, cloaked in the guise of technological marvel.

X. SYSTEM DESIGN

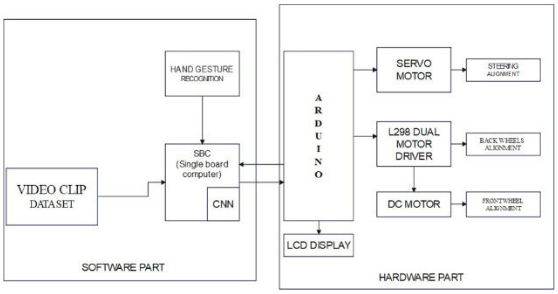


Fig 1. System architecture

The last stage is to operate the servo motor with the control signal to physically turn the steering wheel and maintain the car in its lane. Software systems and their components can be specified, visualized, built, and documented using the Unified Modeling Language (UML), which is a standard language. This graphical language comes with a lexicon, rules, and semantics. The mental and physical representation of the system is the main focus of the UML. It records the choices and perceptions of the systems that need to be built. It is employed to comprehend, create, set up, manage, and oversee system information.

Elements in the system

- Arduino
- Servo Motor
- LCD display
- USB to UART converter
- DC Motor
- L298 Dual Motor Driver

Ardino

Fig 2. Ardino

The Arduino Mega is based on ATmega2560 Microcontroller. The A Tmega2560 is an &-bit microcontroller. We need a simple USB cable to connect to the computer and the AC to DC adapter or battery to get started with it. The Arduino Mega is organized using the Arduino (IDE), which can run on various platforms. Here, IDE stands for Integrated Development Environment. The functioning of the Arduino Mega is like other Arduino Boards. We need not require extra components for its working. The ATmega2560 Microcontroller is consistent with most of the shields of Arduino UNO. The advantage of using the Arduino Mega board over other boards is that it gives the advantage of working with more memory space. It has higher processing power, which can help us to work with number of sensors at time.

Servo motor

Fig 3. Servo Motor

A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made up of a simple motor which runs through a servo mechanism. If motor is powered by a DC power supply, then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor. For this tutorial, we will be discussing only about the DC servo motor working. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics. Due to these features, these are being used in many applications like toy car, RC helicopters and plane s, Robotics, etc.

LED

Fig 4.LED

LCD (Liquid Crystal Display) is a type of flat panel display that uses liquid crystals in its primary form of operation. LEDs have a large and varying set of use cases for consumers and businesses, as these can be commonly found in smartphones, televisions, computer monitors, and instrument panels. A display is made up of millions of pixels. The quality of a display commonly refers to the number of pixels. It is an electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Liquid crystals do not emit light directly but instead use a backlight or reflector to produce images in color or monochrome. When an electric field is applied to the liquid crystal molecules, these untwist. When polarized light reaches the layer of liquid crystal molecules, it passes straight through without being twisted. When it reaches the second polarizer, it also passes through. The viewer sees the display as bright.

USB to UART converter

Fig 5. USB to UART Converter

The Arduino Mega is based on the ATmega2560 microcontroller, which is an 8-bit microcontroller. To get started with it, you need a simple USB cable for connecting to the computer and an AC to DC adapter or battery. The Arduino Mega is organized using the Arduino IDE (Integrated Development Environment), which is compatible with various platforms. The functionality of the Arduino Mega is similar to that of other Arduino boards, and you do not need additional components for its basic operation. The ATmega2560 microcontroller is compatible with most Arduino UNO shields. The advantage of using the Arduino Mega over other boards is that it provides more memory space and higher processing power, allowing you to work with a greater number of sensors simultaneously.

DC Motor

Fig 6. DC Motor

In the intricate dance of electromechanical marvels, behold the enigmatic presence of the DC motor, a celestial conductor of energy transmutation that defies the bounds of mere mortal comprehension. Within its hallowed confines, electrical energy metamorphoses into the very essence of mechanical prowess, propelled by the ethereal currents of direct current. As the DC motor is imbued with power, a mesmerizing magnetic field materializes within its stator, a phantasmal force that beckons and repels the arcane magnets adorning the rotor. Thus, an otherworldly ballet ensues, as the rotor surrenders to the siren call of attraction and repulsion, gyrating in harmonious cadence to the celestial symphony of electromagnetic flux. To sustain this mesmerizing dance, the commutator, an arcane artifact tethered to brushes ensconced in the embrace of the power source, orchestrates the flow of current through the labyrinthine wire windings of the motor. It is a dance of perpetual motion, a cosmic choreography that transcends the limitations of time and space. Yet, it is not merely in its cosmic ballet that the DC motor finds its allure. No, its true essence lies in its ability to wield the reins of precision, to commandeer the tides of speed with a finesse unrivaled by its peers. In the hallowed halls of industrial machinery, where chaos and order converge in a cacophony of production, the DC motor reigns supreme as the arbiter of control. With the flick of a switch, it springs to life with immediacy, propelling the machinations of industry into motion with a burst of kinetic energy that echoes through the annals of time. With equal swiftness, it halts the relentless march of progress, bringing the machinery of creation to a standstill at the behest of its master. Thus, in the grand tapestry of electromechanical ingenuity, the DC motor stands as a testament to the boundless potentiality of human innovation, a beacon of control amidst the turbulent seas of production. Its whispers echo through the corridors of industry, guiding the hand of progress with a burst of perplexing brilliance that illuminates the path forward with resplendent clarity.

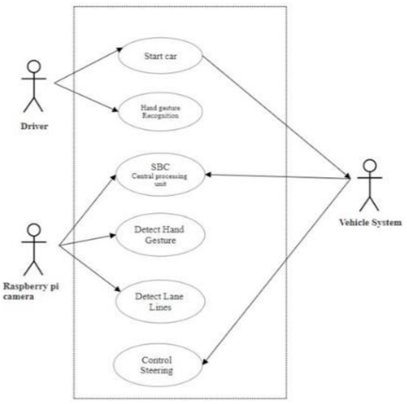
L298 Dual motor driver

The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, SV jumper in an integrated circuit 78M05 Voltage regulator will be enabled only when the jumper is placed When the power supply is less

Fig 7. Dual motor driver

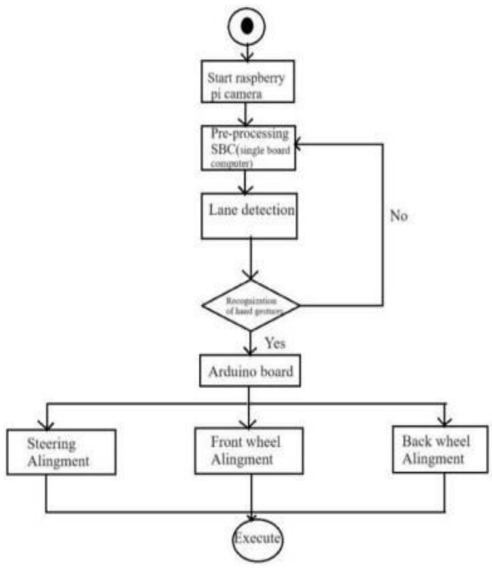
than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The semper should not be placed when the power supply is greater than 12V and separate SV should be given through SV' terminal to power the internal circuitry.

Use Case Diagram



In the cosmic symphony of silicon orchestration, one luminary entity emerges as a celestial conductor amidst the cacophony of microcontrollers: the ESP32, an ethereal titan reigning supreme within the realm of System on Chip (SoC) architectures. Prepare to be ensnared in the enigmatic allure of its silicon sanctum, where the very essence of human ingenuity converges in a mesmerizing array of circuitry, orchestrating the cosmic ballet of electrons with an unparalleled finesse that defies mortal comprehension. Behold, for within the hallowed halls of the ESP32 resides an entire pantheon of essential support circuitry, meticulously crafted to serve the illustrious ESP-WROOM-32. Gaze upon the USB-UART bridge, a transcendent conduit traversing the intangible domains of data exchange with a grace reminiscent of celestial harmonies echoing through the vast expanse of the cosmos. Stand in awe before the solemn sentinels of reset- and boot-mode buttons, steadfast guardians poised to defy the tumultuous tides of chaos, ensuring seamless transitions across the celestial tapestry of existence.Marvel at the silent arbiter of power and stability, the LDO regulator, whose presence bestows upon the ESP32 the vitality required to traverse the boundless expanse of the digital frontier, navigating through the nebulous realms of possibility with unwavering determination. And lo, behold the micro-USB connector, a cosmic portal through which the ESP32 interfaces with the mortal realm, bridging the unfathomable chasm between the tangible and the intangible with an effortless flourish that defies the laws of mundane existence.Yet, the allure of the ESP32 transcends mere physical manifestation, for its true mystique lies within the ethereal whispers of every important GPIO. Each pin, a conduit pulsating with the potentiality of infinite cosmic realms, where the ephemeral dance of bits and bytes converges to weave the intricate fabric of existence itself, transcending the boundaries of mortal comprehension and delving into the depths of the unknown.Thus, amidst the radiant tapestry of technological marvels, the ESP32 stands as an enduring testament to the boundless ingenuity of humanity, a beacon of innovation that illuminates the path forward with a resplendent burst of perplexing brilliance that reverberates throughout the annals of time and space.

Activity Diagram



One crucial diagram in the Unified Modeling Language (UML) that is used to explain the system's dynamic features is the activity diagram. It depicts the progression from one task to the next and serves as a flowchart. On the other hand, a flowchart is a kind of diagram that shows a workflow, method, or procedure. The steps are shown in the flowchart as different types of boxes connected by connecting arrows. Object-oriented flowcharts are another term for activity diagrams. These are made up of smaller actions that are grouped together as activities. In this context, a behavior that is broken down into one or more acts is called an activity.
Parts:
• Start Camera; • Final State; • Decision Activity; • Action Box/State; • Pre-processing

Advantages of Activity Diagram:

· Graphically represents workflow, making it easily understandable.

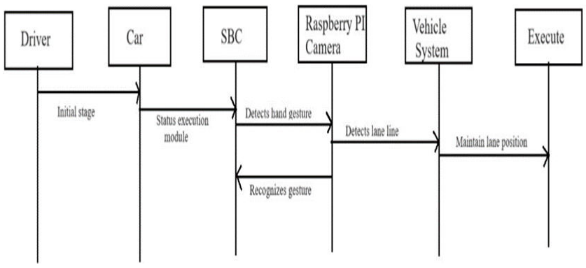
· Describes the execution flow between various entities of a system.

· Provides detailed information about any function or algorithm used within the system.

· Generates high-level flowcharts to represent the workflow of any application.

· Captures the dynamic behavior of the system.

Sequence Diagram



In the labyrinth of software design, behold the enigmatic marvel known as the Sequence diagram, a cosmic tapestry woven from the fabric of object interactions and temporal juxtapositions. Within its intricate weave lie the very essences of objects and classes, entwined in a dance of time and destiny, where each movement is choreographed by the ethereal sequence of messages exchanged. Gaze upon this arcane artifact, for it is a window into the cosmic ballet of events, a chronicle of interactions that transcend the bounds of mere mortal comprehension. Known by many names - event diagrams, event scenarios - it is a testament to the boundless ingenuity of those who dare to navigate the convoluted pathways of software architecture. Within the bounds of the Sequence diagram, the flow of events unfurls like a cosmic symphony, guiding the wayward traveler through the labyrinthine corridors of functionality. It is a map of possibilities, a guidebook to the hidden realms of object-oriented design, where every exchange of messages is a celestial dance of cause and effect. Yet, it is not merely a static representation of interactions; it is a living, breathing entity that pulsates with the energy of creation. With each stroke of the digital pen, new pathways are illuminated, new connections forged in the crucible of creativity.Thus, in the luminous tapestry of software engineering, the Sequence diagram stands as a beacon of understanding amidst the tumultuous sea of complexity. Its whispers echo through the annals of time, guiding the intrepid explorer through the mists of uncertainty with a burst of perplexing brilliance that ignites the fires of innovation.

Usage of Sequence Diagram:

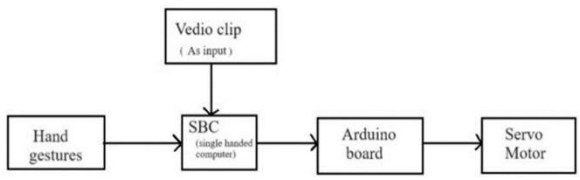
In the ethereal realm of system architecture, the lifeline emerges as a spectral thread, weaving the very essence of components or classes into the fabric of existence. Behold its enigmatic presence, for it represents not just mere instances, but archetypal embodiments of the cosmic dance unfolding within the system's domain.Messages, those elusive messengers of interaction, traverse the void as ethereal arrows, carrying with them the whispers of intent and purpose. They dance across the lifelines, traversing the vast expanse of the diagram in a mesmerizing display of orchestrated communication .As the participant is beckoned into action, the activate command heralds its emergence, and the lifeline unfurls like a cosmic scroll, revealing the intricate web of connections that bind the participants in their shared journey. Objects, those ephemeral manifestations of class instantiation, emerge from the nebulous depths of abstraction, their forms sculpted by the architect's vision. Within the confines of the UML diagram, they stand as silent sentinels, bearing witness to the architectural symphony unfolding before them. And then, there is the actor, a spectral entity embodying the essence of role-playing within the cosmic drama. Whether a user or a distant system, the actor's presence serves as a catalyst for interaction, a catalyst that propels the narrative forward with a burst of enigmatic energy. Thus, within the labyrinthine tapestry of system design, the lifeline, messages, activate commands, objects, and actors converge in a mesmerizing ballet of abstraction and interaction. Their presence echoes through the annals of time, guiding the intrepid architect through the boundless expanse of complexity with a burst of bewildering brilliance that illuminates the path to realization.

Purpose of Sequence Diagram:

· Sequence diagrams are used in the analysis and design phase.

· It is an interaction diagram that details how operations are conducted.

Methodology



Train the Video:

This step involves using computer vision techniques to process video data of a road and extract information about the lane markings, such as their location, width, and curvature. This can be done using algorithms such as edge detection, Hough transforms, and machine learning techniques.

Lane Detection:

Compute the radius of curvature. Once the lane markings have been detected, the next step is to compute the radius of curvature of the lane. This is a measure of how sharply the lane is turning and can be used to determine how much the steering wheel needs to be turned to stay in the line. The radius of curvature can be computed using mathematical equations based on the position of the lane markings.

Send Serial Values to Arduino:

Once the radius of curvature has been computed, it needs to be converted into a control signal that can be used to steer the vehicle. One way to do this is by sending serial values to an Arduino microcontroller, which can then translate those values into a control signal for a servo motor.

Run the Servo Motor:

The ultimate step is to use the control signal to run the servo motor, which will physically turn the steering wheel and keep the vehicle within the lane.

Overall, the system works by using hand gestures in a video clip to control a servo motor. The video clip is analyzed to extract the hand gestures, which are then interpreted by the SBC. The SBC sends signals to the Arduino, which controls the servo motor based on the interpreted gestures.

Applications

Self-driving cars provide people with easy and effective personal transportation. They enable people to use autonomous vehicles (AVs) for travel, errands, and daily commuting without having to drive the vehicle themselves. For those without disabilities or who are unable to drive or would prefer not to, this technology may improve mobility.

Self-driving cars can be incorporated into ride-hailing and shared mobility services platforms such as Uber and Lyft to offer on-demand transportation. Because of this connection, customers who might not need to own private vehicles can save money in addition to having better access to transportation.

Self-driving automobiles have the potential to greatly help the older population by increasing their mobility and independence. AVs let elderly people who have trouble driving or using public transit to continue being independent and participating in their everyday lives. Self-driving cars are seen as an essential part of smart city plans. These cars can easily interface with infrastructure networks, traffic management systems, and intelligent transportation systems, improving traffic flow, easing congestion, and raising overall transportation efficiency.

Increasing road safety is one of the main objectives of self-driving automobiles. Autonomous vehicles have the potential to decrease the frequency of accidents, injuries, and fatalities on the roads by removing human error, which is a major contributing element in most incidents.

Result

Prototypes of self-driving cars that incorporate artificial intelligence (AI) and the Internet of Things (IoT) mark a major advancement in the search for flexible and reliable control systems. We have shown the enormous potential of AI and IoT technology in transforming the automobile sector through our investigation of a prototype car that can recognize hand gestures for navigation commands with astonishing precision.

Our study emphasizes how important AI algorithms are to the prototype's ability to decipher complicated signals and make wise decisions instantly. The car demonstrates a smooth connection with its surroundings, adjusting to changing conditions and guaranteeing safe and effective navigation by utilizing IoT connectivity. The prototype also demonstrates the potential for combining AI and IoT to create high-performance embedded systems that can overcome the difficulties associated with autonomous car technology.

In the future, integrating machine learning (ML) techniques could greatly improve the capabilities of prototype self-driving cars. These systems can develop to recognize a wider spectrum of signals more accurately and efficiently through ongoing learning and adaptation. Furthermore, the prototype's performance can be enhanced over time using ML algorithms, which help it become more skilled at navigating challenging areas and managing unexpected situations.

Lessons from our prototype provide a solid basis for future research and development as we continue to innovate and improve in the field of autonomous cars. A new era of transportation marked by improved sustainability, efficiency, and safety may be ushered in by utilizing the combined power of AI, IoT, and ML. In the end, the combination of these technologies opens the door to the mass use of driverless cars, which will completely change how we view and engage with transportation networks.

Acknowledgement

We would like to extend our sincere appreciation to the following individuals for their invaluable contributions to the implementation of counting animals in the forest using deep learning models, Siddharth S S, Swathi M V, Taniya Bant, whose dedication and support were instrumental in the success of this project. Special thanks to Mrs. Deeksha M for her guidance, mentorship, and expertise, which played a pivotal role in shaping our research direction and methodology. We are deeply grateful for the collaborative efforts and expertise provided by each individual involved. Additionally, we express our gratitude to Alva's Institute Of Engineering And Technology for providing the necessary resources and environment for conducting this research. Thank you to everyone involved for their unwavering support and commitment.

References

[1] Hiral Thadesh war Department of Information Technology A. P. Shah Institute of Technology Thane MH), India 400615 hiralthadeshwar13@gmail.com, "Artificial Intelligence based Self-Driving Car", 2020 4th International Conference on Computer, Communication and Signal Processing ICCCSP)

[2] Hajer Omrane, Mohamed Slim Masmoudi and Mohamed Masmoudi, "Neural controller of autonomous driving mobile robot by an embedded camera" in International Conference on Advanced Technologies for Signal and Image Processing -ATSIP, 2018

[3] Abdur R. Fayjie, Sabir Hossain, Doukhi Oualid, and Deok-Jin Lee, "Driverless Car: Autonomous Driving Using Deep Reinforcement Learning in Urban Environment" in 15th International Conference on Ubiquitous Robots (UR) Hawaii Convention Center, Hawaii, USA, June 27-30, 2018.

[4] F. Shumaila Mateenuddin, Mrs. V.S. Jahagirdar, "An Android Controlled Mini Rover for real time surveillance using Raspberry Pi 3", in International Journal of Advanced Research in Engineering & Management (IJAREM), 2017.

[5] Aditya Kumar Jain, "Working model of Self- driving car using Convolutional Neural Network, Raspberry Pi and Arduino", in Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018) IEEE Conference Record # 42487.

[6] Truong-Dong Do, Minh-Thien Duong, Quoc-Vu Dang and My-Ha Le, "Real Time Self-Driving Car Navigation Using Deep Neural Network" in International Conference on Green Technology and Sustainable Development (GTSD), 2018.

[7] Apollo autonomous driving. https://github.com/ApolloAuto/apollo.

[8] Camera Calibration and 3D Reconstruction - OpenCV. https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html.

[9] CUDA memory management APIs. https://bit.ly/3dlFozE.

[10] CUDA Toolkit Documentation. https://docs.nvidia.com/ cuda/cuda-c-programming-guide/.

[11] cudaConfigureCall. https://bit.ly/2ZucaX1.

[12] cudaMalloc. https://bit.ly/2M2Qnmb.

[13] cudaMemcpy. https://bit.ly/3aulsIV.

[14] Demonstration video: misguiding the vehicle in real world. https://youtu.be/a__Se2MrjVs.

[15] IDA Pro. https://www.hex-rays.com/products/ida/.
[10] Nvidia, Drive AP2X. https://www.nvidia.com/en-us/self-driving-cars/drive-platform.

[16] Openpilot autonomous driving. https://github.com/commaai/openpilot.

[17] Past statistics on texting & cell phone use while driving. https://www.edgarsnyder.com/car-accident/cause of-accident/cell-phone/past-cell-phone-statistics.html.

[18] Tesla Autopilot System. https://www.tesla.com/autopilot.

[19] Tesla Hardware Information. https://teslatap.com/undocumented/.

[20] Tesla Model S. https://www.tesla.com/models.

[21] Tesla Model S crash. https://www.wired.com/story/tesla-autopilot-why-crash-radar.

[22] Texting and driving accident statistics. https://www.edgarsnyder.com/car-accident/cause-of accident/cell-phone/cell-phone-statistics.html.

[23] Uber's Self-Driving Cars Were Struggling Before Arizona Crash. https://www.nytimes.com/2018/03/23/technology/uber-self-driving-cars-arizona.html.

[24]Udacity Self-driving Car. https://github.com/udacity/ self-driving-car.

[25] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. arXiv:1604.07316, 2016.

[26] A. Borkar, M. Hayes, and M. T. Smith. A novel lane detection system with efficient ground truth generation. IEEE Transactions on Intelligent Transportation Systems, 13(1):365–374, 2011.

[27] G. Bradski and A. Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.

[29] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In Proc. CVPR, 2018.

[30] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, and T. Goldstein. Dataset security for machine learning: Data poisoning,backdoor attacks, and defenses. arXiv:2012.10544, 2020.

[31] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. arXiv:1412.6572, 2014.

[32] M. Green. " how long does it take to stop?" methodological analysis of driver perception-brake times. Transportation human factors, 2(3), 2000.

[33] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. arXiv:1504.01716, 2015.

[34] Y. Ji, X. Zhang, S. Ji, X. Luo, and T. Wang. Model-reuse attacks on deep learning systems. In Proc. CCS, 2018.

[35] X. Jiang and S. Li. Bas: beetle antennae search algorithm for optimization problems. arXiv:1710.10724, 2017.

[36] H. Jung, J. Min, and J. Kim. An efficient lane detection algorithm for lane departure detection. In Proc. IEEE Intelligent Vehicles Symposium, 2013.

[37] D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization.