Advancements in Surveillance Robotics: Navigating Motion, Vision, and Perception

Bhanu Teja Giddaluru
bgiddalu@kent.edu
KS811258615

***Abstract*— The publication of "Advancements in Surveillance Robotics: Navigating Motion, Vision, and Perception" is an important milestone in the development of surveillance technology. An innovative combination of an Arduino microcontroller, a Wheels Driver module, and Bluetooth connectivity—all controlled by an easy-to-use smartphone app design by using the MIT app Inventory —produces advanced, mobile-controlled surveillance of a vehicle in this project. Easy Human-Robot Interaction (HRI) is made possible by the intuitive interface, which lets users perform basic commands like drive, reverse, steer, and stop. Live video transmission and a voice assistant powered by ChatGPT are at the heart of this system. They enhance the user experience by providing real-time visual feedback and clever voice command capabilities. Combining cutting-edge software and hardware, this study proves user-centric design works in robotic systems. Our efforts have shown that surveillance robotics may be greatly improved by combining superior mobility with vision capabilities and AI-driven perception. This allows for far more efficient and comprehensive surveillance operations. Future advancements in mobile-controlled robotics will be made possible by this research, which also adds to the present level of surveillance technology.**

Keywords*— *Surveillance, Vision, Motion, HRI, ChatGPT, Bluetooth, MIT app inventory.

I. Introduction

The development of this extremely advanced and multipurpose vehicle system is a giant leap forward in the fields of automated monitoring and computer-aided design. At its core is an extremely sturdy main control board that runs on the well-known and dependable Arduino Mega. The system's brain, the Arduino, orchestrates its many functions and provides limitless room for personalization and expansion. The car's exploratory skills are enhanced and its efficacy in diverse operational contexts is guaranteed by its adaptability.

The automobile uses a Raspberry Pi and an ESP32CAM to combine front and back vision systems innovatively, allowing for full and real-time image capturing. The car's versatility is enhanced by its dual vision arrangement, which opens numerous uses, such as providing improved visual aids and extensive environmental monitoring. A well-engineered battery system, providing both mobility and a considerable power supply, powers this complex array of technologies. An essential component for long-term projects and remote deployments, this guarantees continuous and uninterrupted operations.

The car's capability for many control modalities, each designed to improve the user's engagement and experience, reflects its expert grasp of user-centric design. Various control modalities offer varying degrees of accessibility and convenience, from simple button controls for basic motions to advanced speech recognition for hands-free operation. The ability to use gyroscopes in conjunction with a joystick and gesture controls further demonstrates how precise and versatile the system is.

The ChatGPT Voice Assistant is an interesting new addition to the control environment; it adds AI capabilities like intelligent answers and real-time feedback to improve engagement. Because of its adaptability, the system has many potential uses; for example, it can be used in the classroom to teach students about electronics and programming, in the entertainment industry to provide interactive controls and video functionality, and in security, applications to allow for remote monitoring and area observation. Design and user experience were meticulously considered in the project's user interface. The easy interface, created using MIT App Inventor, lets any demographic drive the car. This focus on usability improves technological accessibility and stimulates user invention and experimentation.

In addition, the car's live streaming technology improves real-time remote monitoring. Front and rear vision systems provide live feeds for security surveillance and remote help. This feature enhances usefulness and expands telematics. This technology emphasizes the project's commitment to creating a flexible tool that meets users' changing needs.

In conclusion, ChatGPT as a voice assistant enhances human-robot interaction in the project. Complex command processing and a natural language conversational interface are possible with this new method. It marks a shift toward intelligent automation where the car can execute duties based on voice commands, making it a smart companion rather than merely a tool. Harmonizing this advanced AI with the car's control systems shows the project's forward-thinking approach, combining cutting-edge technology with practical utility.

The remainder of this paper is organized as follows. I begin in Section 2 providing a Literature Review, Problems and Challenges in Section 3. Project design in Section 4 Implementation in Section 5, Evaluation plan in Section 6 Section 7 Results, Discussion in Section 8, Conclusion Section 9, and References in Section 10.

II. Literature Review

This paper "Voice Controlled Robotic Car Using Mobile Application" examines a voice-controlled robotic car for paralyzed people. Human-robot interaction (HRI) and Android smartphone voice commands drive the robotic automobile. This interface improves independence and quality of life for physically disabled individuals by making it easier to handle items like wheelchairs and automated systems. An Arduino Mega 2560 microprocessor, infrared sensors, a Bluetooth module, DC motors, and a chassis make up the vehicle's architecture.

Fig. 1. Interface for AMR_Voice.

Speech recognition on the smartphone app converts spoken commands into Bluetooth-transmitted data for the automobile. The automobile can drive itself and avoid obstacles thanks to the instructions processed by the Arduino board. This project showcases a method for operating robotic cars using voice commands and Bluetooth technology. It also suggests ways to improve this method, such as adding GPS integration to increase the communication range.

This study "Development of Single-board Computer-based Self-Driving Car Model using CNN-Controlled RC Car" prototypes a small-scale autonomous car using an RC car and a single-board computer to implement and test a convolutional neural network AI model. The model predicts steering angles and car speeds using camera footage. Different training data (3000, 6000, 12000, 24000) were used to train models with variable convolution layer counts (2, 3, 4, 5).

Predicting steering angles in different scenarios assessed the prototype's performance. At this point, I put the automobile through its paces by seeing how well it could use the trained CNN models to anticipate turns and speeds. I wanted to see how well the car could drive itself around a track while responding to different kinds of environmental signals that it might detect with its camera. The mean absolute deviation from the actual values of the steering angles predicted by the AI model served as a measure of its performance. The model with 24000 training data and three convolution layers got the lowest absolute prediction error. This study outlines the potential for autonomous vehicle development, recommending advanced deep learning structures and additional sensors such as ultrasonic and LiDAR for enhanced performance.

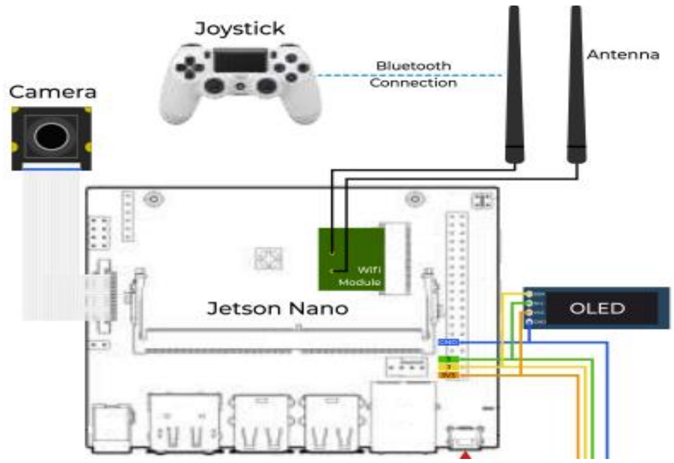

Fig. 2. Diagram of Self-Driving Car

The paper, "IoT Application Development Using MIT App Inventor to Collect and Analyze Sensor Data" explains how the MIT App Inventor now has Internet of Things (IoT) features built-in, making it possible to build IoT apps with an intuitive UI. This opens the door for data collecting and analysis by letting both seasoned programmers and those just starting to build apps for platforms like Android Things and Raspberry Pi. The study emphasizes the importance of easily accessible development platforms and the expansion of the Internet of Things (IoT) area.

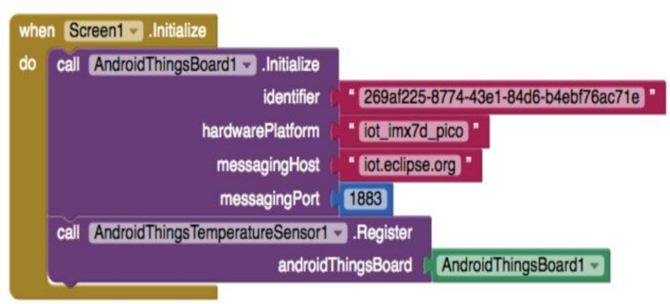

Fig. 3. Diagram of MIT App Inventor's program block

Connecting Internet of Things (IoT) devices to mobile apps is made possible by the integration's components, which include the Android Things Board and sensor extensions. The article uses a temperature monitoring app to demonstrate this point, which makes use of thresholds to dynamically alter the colour of the phone's screen in response to measured temperatures. The potential for more people to innovate in this industry may be democratized by focusing on the possibility in the conclusion, which is about the development of IoT applications.

The paper "Location Sensitive Browser App using MIT App Inventor for Blocking Social Media Websites in Educational Institutions" introduces a location-sensitive mobile browser that blocks social media websites in classrooms and auditoriums to reduce student and teacher distractions. The MIT program Inventor-developed browser program identifies the user's location and modifies website accessibility to offer educational materials while banning Facebook, WhatsApp, and YouTube in specific locations. This design addresses the need for regulated internet access in educational environments, as unrestricted access to specific websites might hinder learning and lead to non-productive activities.

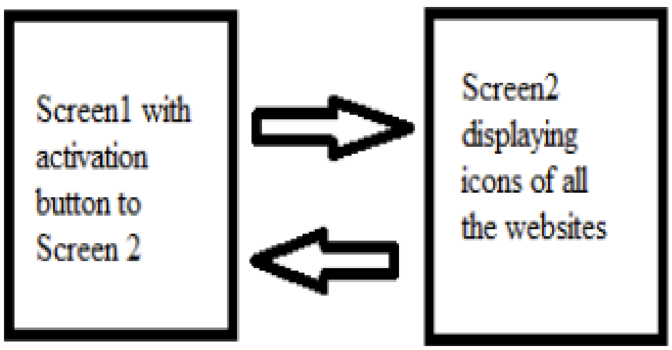

Fig. 4. Interfaces accesses.

It highlights MIT App Inventor's user-friendly, cloud-based visual programming environment for iOS and Android app development. Quick development with minimal programming knowledge is possible with this strategy. The app's GUI and backend programming make navigation easy and control website access by location. The app worked well in university buildings, with the location sensor being reliable in most regions but limited at building borders. The report advises employing Wi-Fi access points for more precise position tracking in the future.

In this paper titled "Design and Implementation of an Autonomous Indoor Surveillance Robot based on Raspberry Pi," the main objective is to create a budget-friendly robot that can autonomously monitor indoor spaces for security reasons. Designed to make regular patrols inside, the robot uses several sensors to identify unusual activity, such as the presence of people or changes in temperature. Thanks to its adaptable design, it can do manual control from a distance and incorporate new software features with ease.

Utilizing a Raspbian OS and the ROS framework, the robot can function independently. It employs a traction belt system and DC geared motors for mechanical navigation across various terrains, and it interacts with its environment using a variety of sensors, including lidar, gyro, accelerometer, and temperature. To lessen the computational burden on the robot, high-level tasks like face recognition are handled by a remote server, and ROS is equipped with software components written in Python and C++.

III. Problems and Challenges

I face software challenges. Developing the software for our surveillance vehicle was fraught with tricky obstacles. A solid grasp of motion mapping was primarily required for the MIT App Inventor application development and environment setup processes. Accurately controlling the vehicle's pitch and roll requires us to tackle the challenging task of coordinating these sensors with the vehicle's motion. Further complications arose from the need to incorporate speech recognition for operation via voice. It was our responsibility to make sure the system could comprehend and act upon a wide range of voice commands in real-time. Integrating ChatGPT into our software was probably the most daring part.

Extensive testing and careful coding were necessary to guarantee that this integration would work as intended, but it was essential for improving user interaction. Plus, for remote surveillance to work, a reliable and high-quality live video feed has to be set up. To improve the vehicle's capabilities, this feature improved the operator's visual input in real-time, but it required a lot of work to provide a stable network and efficient data transfer.

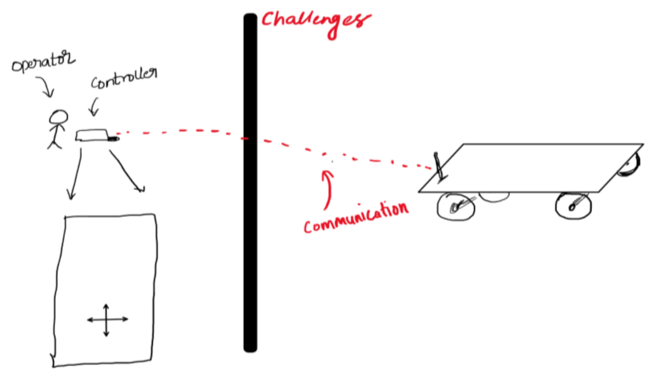

Fig. 5. Challenges in communication.

Difficulty with Vision Integration and Remote Vehicle Control: I had a lot of trouble with enabling the driver to control the vehicle from a distance, even though they were in separate places. For this part of the project, a reliable communication system that could send commands across great distances without any delay or signal loss was required. This obstacle included the incorporation of live video transmission. Efficient surveillance needed to have a high-definition video feed that could be accessed in real-time by the operator.

In addition to integrating high-tech camera systems, this mission also required optimizing data streaming over networks that could be unstable. There was already a lot of complexity without having to worry about the video feed having to be clear and fast enough to coordinate with the vehicle's controls and movements.

Problems with Power Distribution and Firmware Uploading I encountered numerous problems with the hardware, the most significant of which were faults in the distribution of power and firmware uploads. Problems with compatibility and connectivity made it a trial-and-error procedure to upload the right firmware to the ESP32 and Arduino boards. Fixing these mistakes was a major pain, but they were essential for the car to work properly. I also had to pay close attention to how the system's power was distributed. It was a fine line to walk to avoid overloading the system while still giving each part the power it needed.

Particularly difficult was this task because the various components, including motors, sensors, and communication modules, had different power requirements. To keep operations stable and avoid system failures that could cause the vehicle to lose control or get damaged, an effective power management approach was necessary.

IV. Project Design

An Arduino-powered robotic vehicle that can be programmed using a smartphone app is at the core of this project. The vision capabilities of the system are enabled by a Raspberry Pi for real-time image processing and an ESP32 camera module. With the integration of ChatGPT, voice-operated controls are made possible through natural language understanding. Improvements to the design and connectivity are obtained by feedback loops, guaranteeing a smooth user experience. Combining mechanical control, cognitive interaction, and superior vision processing, these technologies form a hybrid system that has numerous robotic applications.

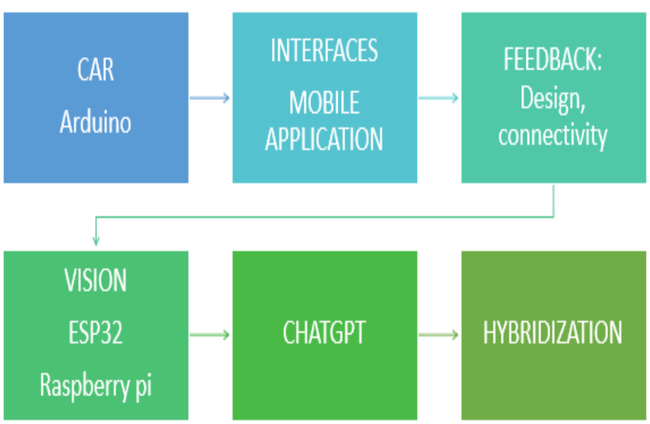

Fig. 6. Images of driver corresponding to Yawn and No_Yawn.

4.1. INTERFACE DESIGN

The car and user communicate via the mobile app. Arduino commands are communicated to the car via the control panel. Remote automobile control is possible via Bluetooth between the app and Arduino.

Fig. 7. Different Interface control methods

4.1.1. BUTTON CONTROL

The "Button Control" uses a Bluetooth-connected smartphone app for easy car control. Five buttons control fundamental movements: Forward, Backward, Left, Right, and Stop. The app's buttons make these controls easy to use, and a HOME button returns to the main screen, making car operating simple and efficient.

4.1.2. VOICE CONTROL

Through a Bluetooth-enabled mobile app, "Voice Control" offers hands-free car operation. There are two buttons: one for voice command mode and one for "Speak". The car may be controlled by saying "Front," "Back," "Turn Left," "Turn Right," and "Stop". Combining vocal contact with system control dynamics improves user accessibility and convenience. Also included is a HOME button for fast return to the app's main screen after command execution.

4.1.3. JOYSTICK CONTROL

Through the use of a Bluetooth connection, the "Joystick Control" functions offer a user-friendly method to steer the vehicle. The user can steer the automobile in any direction, reverse its direction, turn left or right, and even stop it all by using a virtual joystick that is part of the smartphone app. By emulating the feel and operation of classic gaming controllers, this strategy makes driving a more natural and natural experience. Users can easily return to the app's main interface after using it by pressing the HOME button.

4.1.4. GESTURE CONTROL

An exciting new way to control the vehicle with your body is here with the "Gesture Control" feature. As soon as a Bluetooth connection is set up, users can order the car to go ahead, backwards, or turn in any direction by performing gestures. With this strategy, the control system becomes more immersive, allowing for a more hands-on approach to navigating the car. With the addition of a HOME button, users can quickly return to the application's main menu after controlling the car.

4.2. VISION INTEGRATION

Fig. 8. Images of driver corresponding to Yawn and No_Yawn.

4.2.1. FRONT CAMERA

A Raspberry Pi, selected for its powerful processing capabilities—essential for handling complicated data from the front-facing webcam—forms the clever backbone of the front vision system. A high-resolution picture of the car's immediate surroundings is provided by the webcam, which is strategically positioned at the front of the vehicle and collects both still photographs and films.

This configuration enables the execution of complex image processing algorithms on the Raspberry Pi, which is essential for navigation and obstacle detection. The incorporation of this technology permits analysis and response in real-time, which is essential for the car's autonomous capabilities and for giving the user a live visual feed.

4.2.2. RARE CAMERA

The ESP32 module is used in the rear vision system because of its small size and built-in Wi-Fi capabilities. This allows the visual data to be transmitted back to the main control system without the need for tedious wiring. Because of its integrated camera interface and low power consumption, the ESP32 camera module is perfect for this application; it captures the vehicle's back view in a simplified and economical manner.

In addition to improving spatial awareness, this rear camera module boosts the vehicle's safety features by keeping an eye out for potential impediments and helps with reverse operations.

4.3. CHATGPT INTEGRATION

One important part of the project is ChatGPT integration, which allows for natural language processing capabilities and improves the user experience. By integrating ChatGPT through application programming interfaces, consumers can have natural-sounding conversations with their vehicles. The integration of ChatGPT with the MIT App Inventor-built mobile app makes this possible. Users can ask inquiries, get status updates, and give voice instructions all through an easy-to-use interface made possible by the connection. The addition of an AI-driven communication layer greatly improves the vehicle's accessibility and user-friendliness.

4.4. HYBRIDIZATION

At the hybridization stage, all the project's subsystems come together to form a single, unified whole. It integrates the mechanical control of the Arduino with the dynamic input methods of the Raspberry Pi and the ESP32 module, the advanced vision processing of the buttons, voices, joysticks, and gestures, and the intelligent interaction of ChatGPT. This comprehensive integration guarantees that all parts work together to provide a smooth operating experience.

It showcases the project's interdisciplinary approach, which combines software, hardware, and artificial intelligence to build an intelligent robotic vehicle that can adapt to different settings and carry out different duties.

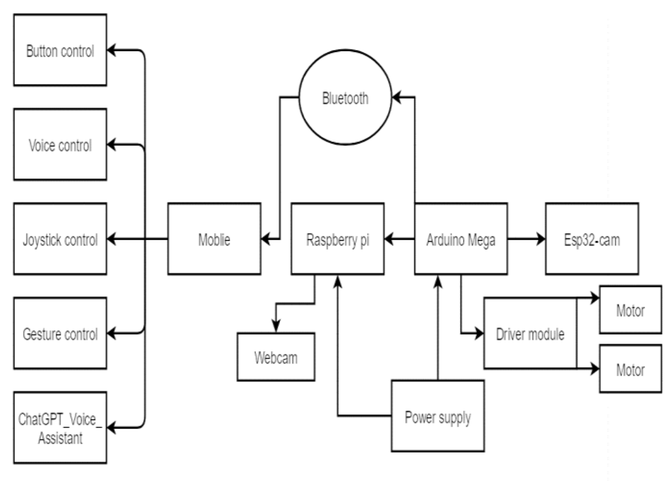

Fig. 9. Block diagram of hybridization

The block diagram is a representation of the whole design of a system for controlling a robotic vehicle. Connected via Bluetooth to a mobile app, the system incorporates several inputs for control, including as buttons, speech, joystick, gesture controls, and a ChatGPT voice assistant. With the help of the Arduino Mega and the ESP32-cam, the Raspberry Pi acts as a central processing hub and allows for precise control of the motors via a driver module.

With a dependable power source supporting the design, this layout allows for various user interaction modes and supports the car's front and rear vision capabilities. A high level of human-robot interaction is embodied in the system's architecture, which is designed to be sensitive and flexible.

V. IMPLEMENTATION

In the paper about the surveillance robotic car's implementation, the following sections describe the interface's design and functionality, as well as the integration of different control and visual systems.

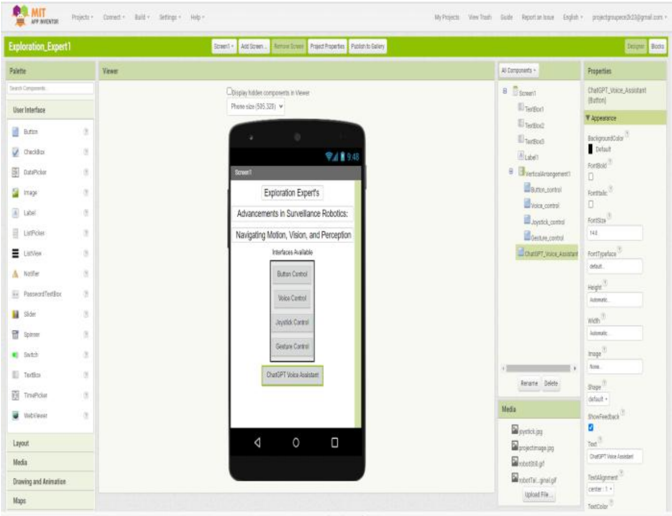

Fig. 10. MIP app Inventor for Interface Design

I chose MIT App Inventor for its flexible environment, user-friendly design, and powerful features, and I started developing our mobile app there. With its intuitive interface and built-in voice, joystick, and gesture controls, this app is perfect for controlling our robotic surveillance system. It also integrates with the popular ChatGPT voice assistant.

Making sure the app's UI was easy to understand and use meant that anyone with varying degrees of expertise could manage the robotic car with no problem. One of the most important steps in implementing our project successfully was using this platform to quickly prototype and test several versions of the user experience.

5.1. INTERFACE DESIGN

A smooth and easy-to-understand experience for the user was the primary goal in developing the interface. A mobile app was developed using MIT App Inventor. It has an organized design with responsive buttons for each control mode. Users without a technological background will have no trouble navigating and controlling the robotic car because to its intuitive design.

5.1.1. BUTTON CONTROL

Five buttons—Forward, Backward, Left, Right, and Stop—represented the fundamental movements in the button control implementation. For navigational activities that demand accuracy and quick reflexes, this control's simplicity is essential since it provides immediate and exact responses to user inputs.

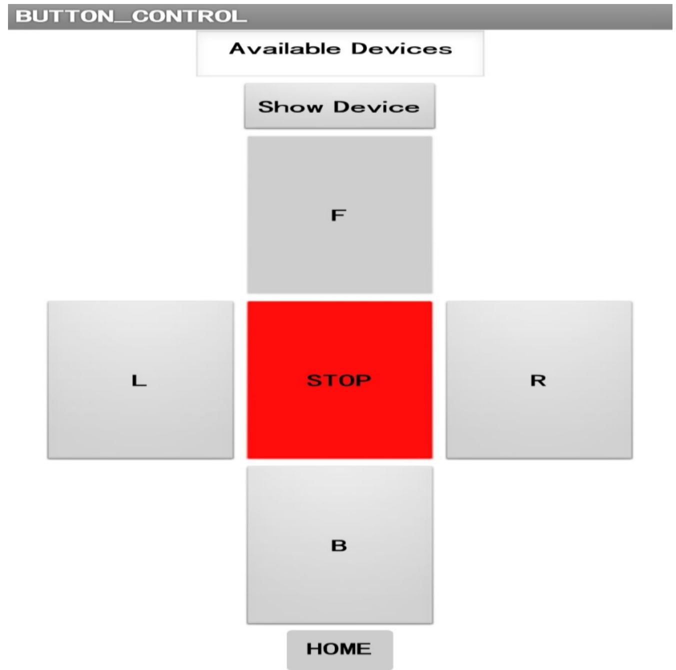

Fig. 11. Interface for button control

5.1.2. VOICE CONTROL

An option for hands-free operation was added using voice control, making it more accessible. With this function, customers can control the vehicle with their voice thanks to speech recognition technology. When using hands-on control would be too risky or impractical, this method comes in handy.

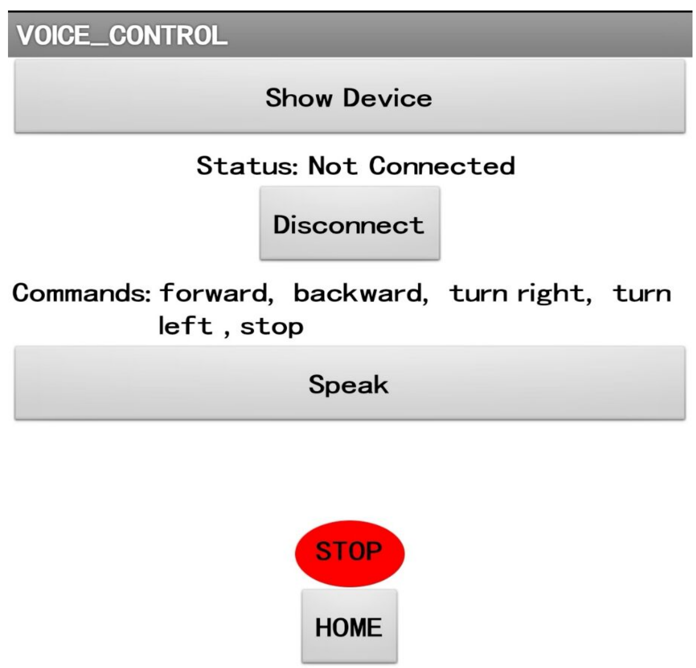

Fig. 12. Interface for voice control

5.1.3. JOYSTICK CONTROL

Joystick Control A fun and comfortable method to drive is using a joystick, which mimics the feel of a classic steering wheel and other manual controls. The app digitally achieves analogue-style control, giving users a dynamic and precise driving experience.

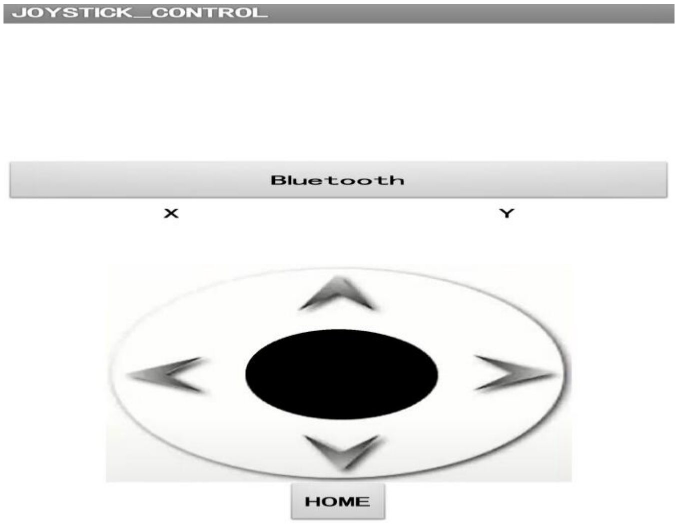

Fig. 13. Interface for joystick control

5.1.4. GESTURE CONTROL

Control by Gesture With gesture control, the user can tell the car where to go by tilting and rotating their mobile device, which the sensors in the device read as commands. A more engaging and entertaining user experience is provided by this novel method.

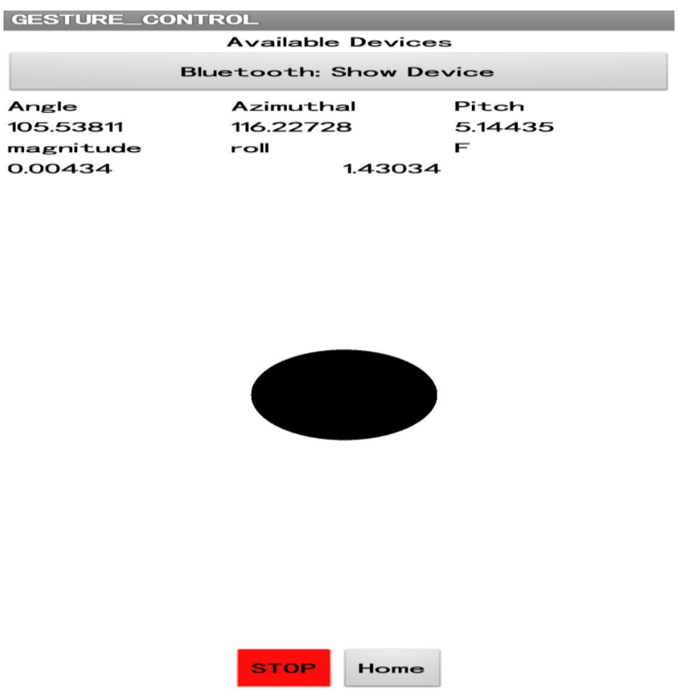

Fig. 14. Interface for gesture control

5.2. VISION INTEGRATION

To enable live streaming, the front and back vision systems—powered by Raspberry Pi and ESP32-CAM, respectively—were integrated. With the ESP32-CAM transmitting the rearview and the Raspberry Pi processing the footage from the front webcam, the driver can see more of their environment, improving their situational awareness and surveillance skills.

Fig. 15. Front Camera with Raspberry Pi.

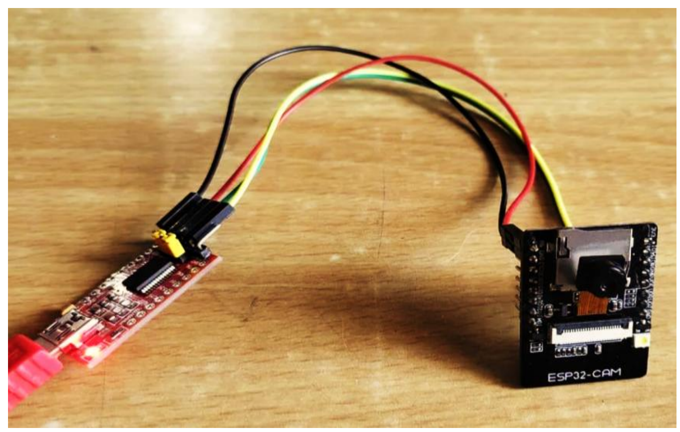

Fig. 16. Rare camera with ESP32

5.3. CHATGPT INTEGRATION

Integration of CHATGPT The automobile was able to comprehend and react to intricate requests and commands because ChatGPT was incorporated into the system through an API. This AI-powered method improves the experience by making the control system more advanced and by allowing the user to have a conversation with the car.

5.4. HYBRIDIZATION

Hybridization of all control methods, vision systems, and ChatGPT integration culminated in the project, providing a responsive and multifunctional surveillance car. To provide a strong and versatile platform that can handle different tasks in different situations, this hybrid system uses the qualities of each component to work together.

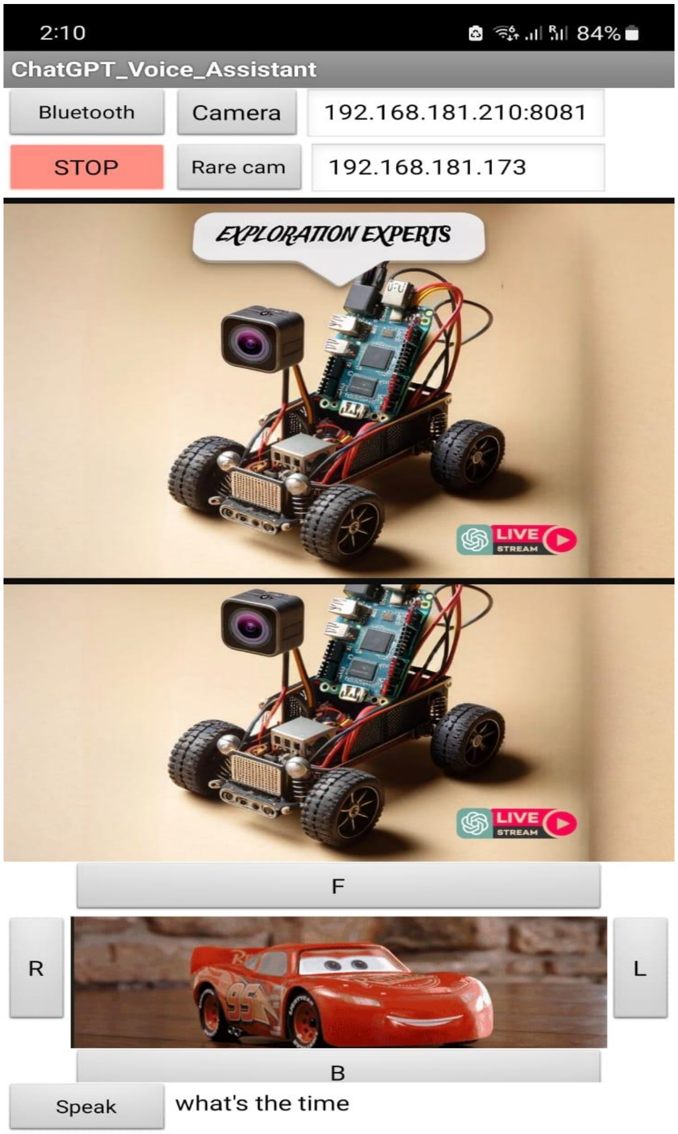

Fig. 17. Hybridization of control methods mobile controll application.

Hybridization Application the Interface with the name ChatGPT Voice Assistant, it contains the Bluetooth button to manual connectivity to the car, and the camera button are used to connect the Front and Rare camera of the car. For live video streaming enter the IP address for both cameras and it will be connected automatically and displayed in the 2 boxes in the application.

In the first box, the Front cam video live streaming and the bottom for the rare car video streaming. It also has 5 buttons for the manual car operation. The speak button is connected to the ChatGPT the user input will be displayed in the side response. The ChatGPT is accessed using the API Key, and the response as the Output speak and an animated talk of the Car present in between the button interface. In the hybridisation, it contains the 2 cameras' live video transmission, along with the integration of the ChatGPT making the Application more efficient and user-friendly.

VI. Evaluation Plan

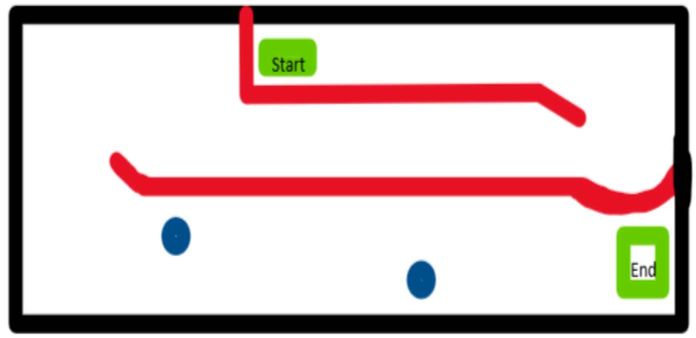

Fig. 18. Test bench for calculating the evaluation plan.

Define specific Key Performance Indicators to measure the success and effectiveness of the implemented approaches. KPIs should be quantifiable and directly aligned with the study's objectives.

6.1. Quantitative Evaluation Metrics

Metric 1: Communication Efficiency: Measure the speed and reliability of communication between Arduino Mega, Raspberry Pi, and ESP32CAM. Use metrics such as data transfer rates and latency.

Metric 2: Computational Speed: Evaluate the computational speed and efficiency of tasks performed by the integrated system. Measure processing times for complex algorithms and tasks.

Metric 3: Vision System Accuracy: Assess the accuracy of the vision system in real-time image capture and processing. Use metrics like image recognition accuracy and response time.

6.2. Qualitative Evaluation Measures

Measure 1: User Satisfaction: Conduct surveys and interviews to gather feedback on the user experience. Assess users' satisfaction with the system's usability and performance.

Measure 2: System Reliability: Evaluate the system's reliability in different scenarios through user feedback and observations. Identify any instances of system failures or inconsistencies.

6.3. Comparative Analysis

Analysis 1: Comparative Study with Existing Models. Compare the performance of the developed integration framework with existing models in similar domains. Highlight advantages and disadvantages in comparison.

Analysis 2: Performance Before and After Optimization. Compare system performance metrics before and after the implementation of optimization strategies. Assess the impact of optimization on communication, computation, and vision system capabilities.

6.4. User Testing and Feedback

Testing 1: Usability Testing: Conduct usability testing with end-users to evaluate the system's ease of use. Gather feedback on user interaction and system navigation.

Testing 2: Real-world Testing Scenarios: Implement the integrated system in real-world scenarios. Evaluate the system's performance, adaptability, and robustness in diverse environments.

Data Collection Procedures: Detail the procedures for collecting both quantitative and qualitative data, including the tools and methods used for data gathering.

Data Analysis Approach: Specify the approach for analyzing collected data, including statistical methods for quantitative data and thematic analysis for qualitative data.

Reporting and Presentation: Outline how the evaluation results will be presented, including reports, visualizations, and any supplementary materials.

VII. RESULTS

7.1. Comparison of Control Methods

7.2.1. Button Control

Our evaluation of button control revealed that it is a straightforward and responsive method for commanding the surveillance vehicle. Users reported a high level of satisfaction with its simplicity and reliability. The button control scheme exhibited minimal latency, making it suitable for quick and precise commands.

7.2.2. Voice Control

Voice control demonstrated effective integration with the system, allowing users to interact naturally with the surveillance vehicle. Users praised the hands-free nature of voice control, emphasizing its convenience in scenarios where manual input might be challenging.

7.2.3. Joystick Control

Our assessment of joystick/controller control highlighted its versatility, providing users with fine-grained control over the surveillance vehicle's movements. The sensitivity of the joystick was optimized for precise manoeuvres, earning positive feedback from users. The ergonomic design of the controller contributed to an enhanced overall user experience.

7.2.4. Gesture Control

Gesture control, although innovative, exhibited some challenges in terms of accuracy. While it performed well in controlled environments, its effectiveness was slightly diminished in complex scenarios. Future refinements to the gesture recognition algorithm are recommended to improve its robustness across diverse settings.

7.2.5. ChatGPT Voice Assistant

The integration of ChatGPT's voice assistant greatly enriched the user experience. Users appreciated the natural language interaction, with the voice assistant accurately interpreting and executing a variety of commands. The context-aware nature of the voice assistant contributed to fluid conversations and improved overall system usability.

7.2. Video Transmission Effects

7.2.1. Real-time Performance

Video transmission displayed commendable real-time performance. Different network conditions, including Wi-Fi were tested, and the system adapted seamlessly to fluctuations, maintaining reliable video feeds under various circumstances.

7.2.2. Clarity

The clarity of video transmission remained high across different transmission methods. Even in bandwidth-constrained environments, the system ensured a clear and detailed video stream. Users could rely on the transmitted video for accurate situational awareness, crucial for effective surveillance and control.

7.2.3. Algorithm Accuracy

Our chosen object detection algorithm demonstrated a high accuracy rate, correctly identifying objects in the surveillance environment. False positives were minimal, contributing to the overall reliability of the object detection system.

7.2.4. Real-time Performance

The object detection algorithm exhibited real-time performance, seamlessly integrating with the video transmission. Objects were identified and tracked in near real-time, providing operators with timely information for decision-making. Performance tests under varying hardware conditions confirmed the adaptability of the algorithm.

Fig. 19. Object detection using rare camera

VIII. Discussion

Arduino can interface with a variety of sensors such as cameras, accelerometers, gyroscopes, and distance sensors. Real-time control systems can process data from these sensors to make decisions or adjustments in real-time. Cameras interfaced with Raspberry Pi and ESP32-CAM can capture live video feeds. They can process and analyze video data in real time.

ESP32-CAM can perform basic image processing tasks such as edge detection, object recognition, or colour tracking. Integration with real-time control systems allows for dynamic adjustments based on the analysis of live video feedback. For user-friendly interaction, Arduino can be connected to a computer or mobile device to display real-time data and receive user inputs through a graphical interface. Live video feedback can be displayed on a screen alongside control parameters.

Integrate ChatGPT with a robust speech recognition system to convert spoken language into text. Performance Improvement: Choose a high-quality speech recognition API or engine that can handle various accents, languages, and background noises. This ensures accurate command recognition. Optimize the voice assistant for real-time processing to minimize latency. Performance Improvement: Use efficient algorithms and hardware to handle real-time speech recognition and generation, ensuring prompt responses to user commands.

IX. Conclusion

This project highlights the unique integration of technology and user-focused design in surveillance robotics. A multi-functional robotic automobile system with front and rear vision enabled by Raspberry Pi and ESP32-CAM is a milestone in autonomous vehicle design. Real-time monitoring is practical, informative, and entertaining with this technology. Efficient button, voice, joystick, and gesture controls integrated into a user-friendly mobile interface support the project's goal of improving human-robot interaction.

The voice assistant and live video broadcasting features of ChatGPT play a crucial role in command and control. The voice assistant enables users to interact with the surveillance vehicle through natural language, while live video broadcasting provides users with real-time visual feedback, strengthening the monitoring and decision-making processes.

The resilience of the live video feedback system is crucial for real-time surveillance and remote monitoring. This ensures that the surveillance vehicle can consistently provide clear and reliable video streams in complex environments, enabling operators to make accurate decisions. The project inspires future field advancements. The effective hybridization of varied control systems with sophisticated vision processing and AI interaction shows what robotics, AI, and user-centric design can do. Project successes lay the path for future research and development, promising improvements in autonomy, navigation precision, and AI feature integration. After completion, the project offers a legacy of invention and a template for future surveillance robot pioneers.

X. References

[1] M. S. Hamzah, N. Fadillah, D. W. Maulana, I. M. Joni, C. Panatarani and F. Faizal, "Development of Single-board Computer-based Self-Driving Car Model using CNN-Controlled RC Car," 2022 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2022, pp. 1805-1812, doi: 10.1109/ICEARS53579.2022.9751873.

[2] K. Akila, B. Sabitha, A. Shanmugavadivel, M. Praveen and N. H. Akbar Ali, "Tele-Operated `4W Vehicle," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 2021, pp. 1-5, doi: 10.1109/ICAECA52838.2021.9675769.

[3] S. Gholami, V. R. Garate, E. De Momi and A. Ajoudani, "A Shared-Autonomy Approach to Goal Detection and Navigation Control of Mobile Collaborative Robots," 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), Naples, Italy, 2020, pp. 1026-1032, doi: 10.1109/RO-MAN47096.2020.9223583.

[4] S. Chakraborty, N. De, D. Marak, M. Borah, S. Paul and V. Majhi, "Voice Controlled Robotic Car Using Mobile Application," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, India, 2021, pp. 1-5, doi: 10.1109/ISPCC53510.2021.9609396.

[5] T. Dharmasena and P. Abeygunawardhana, "Design and Implementation of an Autonomous Indoor Surveillance Robot based on Raspberry Pi," 2019 International Conference on Advancements in Computing (ICAC), Malabe, Sri Lanka, 2019, pp. 244-248, doi: 10.1109/ICAC49085.2019.9103399.

[6] G. R. de Campos, F. D. Rossa and A. Colombo, "Safety Verification Methods for Human-Driven Vehicles at Traffic Intersections: Optimal Driver-Adaptive Supervisory Control," in IEEE Transactions on Human-Machine Systems, vol. 48, no. 1, pp. 72-84, Feb. 2018, doi: 10.1109/THMS.2017.2776205.

[7] C. Huang, F. Naghdy, H. Du and H. Huang, "Shared control of highly automated vehicles using steer-by-wire systems," in IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 2, pp. 410-423, March 2019, doi: 10.1109/JAS.2019.1911384.

[8] https://ieeexplore.ieee.org/abstract/document/8457076?casa_token=EK7oEWGS35wAAAAA:YCzZOn3IaqekUiqu9-gDqqLTOTIiusv63CxRLp_MRLyP_CSUIm3ycoK-8kMHo8yPqlr5m8tH

[9] https://ieeexplore.ieee.org/abstract/document/7451738?casa_token=3UOgEG_mzc4AAAAA:tN0E6DEU8HkgCSQ3nFmYdb4stCR3YlfPFLIq-wQOP-AYmIgaZcis0GcUWVbXmjWvcD7hrHwH

[10] https://ieeexplore.ieee.org/abstract/document/8202319?casa_token=RiGsJFrr6lUAAAAA:J3UnAUmm6b_SeRs9CYPB0AMvui8t7EKLSjuW8BRzLTJS1XGKt_nAR1FYamYC0E7VjoQO1tDe