VIoT: A Voice and IoT Controlled Unmanned Automated Vehicle

(Blinded for Review)

***Abstract*—VIoT, a Voice and IoT Controlled Unmanned Automated Vehicle (UAV), pioneers a blend of Voice, IoT, and Bluetooth technologies to redefine remote-controlled robotics. Powered by Rock Pi 3a and an Android app, users seamlessly control the car using Bengali voice commands transmitted through IoT for long-range communication. It also uses Bluetooth for short-range operations if the vehicle is within approximately 20 meters of range from the controlling app. The dual-mode control mechanism offers flexibility, allowing users to steer the robotic car through the Android app effortlessly. Leveraging Google Speech Recognition API for Bengali voice input, this paper establishes a new standard for user-friendly robotic interfaces. Using WebSocket instead of an actual database like Firebase as the medium of data passing for IoT communications significantly solves latency problems. The comprehensive hardware setup includes Rock Pi 3a, L293D motor driver, and basic components, setting a milestone in human-robot interaction and expanding adaptability across various applications.**

Keywords*— IoT, Remote-controlled Vehicle, Bengali Voice Commands, Unmanned Automated Vehicle

I. INTRODUCTION

Traditional robotic control systems often struggle to provide intuitive interfaces for user interaction, leading to fragmented functionalities and limited control options. Additionally, existing systems may lack support for diverse control methods, such as voice commands and remote controller inputs. This paper aims to bridge these gaps by developing a unified platform that empowers users with flexible control options, including voice commands via a mobile application and remote controller inputs. Also, the need for different systems in short range and long range is a key issue to be considered in terms of a complete robotic system. By addressing these challenges, we strive to enhance the user experience and pave the way for more intuitive and inclusive human-robot interaction.

While existing UAV researches demonstrate notable advancements in areas such as surveillance, voice control, and human-robot interaction, there remains room for further development and enhancement. This paper aims to address the limitations and challenges present in these existing researches by introducing innovative solutions and improvements. The surveillance robocar utilizing IoT and the Blynk app [1], while effective for surveillance purposes, may encounter limitations in its control methods. Voice-controlled autonomous vehicles through Google Assistant [2] and robotic cars using mobile applications [3] also face hurdles related to the accuracy and reliability of voice recognition systems, especially in noisy environments or with varying accents. Additionally, the human voice-controlled robotic car [4] research, although aiming for simplicity, may overlook scalability and robustness in its design for broader applications. Addressing these issues, this paper seeks to enhance the control methods, improve voice recognition accuracy, and ensure scalability and robustness in the design to overcome the limitations observed in existing researches.

This paper revolves around the need for a robust and versatile robotic control system that seamlessly integrates various control methods while ensuring efficient communication between the user interface and the robotic hardware. The methodology employed in this paper contributes significantly to the advancement of robotic car technology by addressing several key challenges. Firstly, the integration of advanced hardware components, including the Rock Pi 3a Micro-controller and L293D Motor Driver, ensures precise control and efficient operation of the robotic car. Secondly, the utilization of Python scripts on the Debian OS enhances the software layer, enabling seamless data retrieval, GPIO signal translation, and communication management for both IoT and Bluetooth functionalities. Lastly, the implementation of multi-threading and parallel execution techniques optimizes data processing and enhances system performance, thereby improving overall responsiveness and user experience. Collectively, these methodological contributions enhance the functionality, reliability, and efficiency of the robotic car system, paving the way for its broader application and adoption in various domains. It not only advances the hardware and software components but also revolutionizes the user interaction paradigm through voice and remote-control capabilities. By integrating voice recognition technology, the robotic car becomes accessible to users through natural language commands, enhancing usability and accessibility. Additionally, the incorporation of remote-control functionality allows users to operate the robotic car providing flexibility and convenience in various scenarios. Through these innovations, the methodology not only improves the technical capabilities of the robotic car but also enhances user engagement and interaction, opening up new possibilities for its practical applications in surveillance, exploration, and beyond.

The rest of the paper is structured into 4 main sections: Related Works, System Methodology, Experimental Analysis, and Conclusions. In the Related Work, existing literatures relevant to this research are reviewed, highlighting their methodologies and findings. Section 3, The System Methodology outlines the approach taken in the study, detailing the hardware components, software architecture, and the implementation of voice control and remote-control functionalities. Following this, section 4, the Experimental Analysis, presents the results of experiments conducted to evaluate the performance and effectiveness of the developed system. Finally, section 5 concludes the paper with the key findings of the study, its implications, and potential avenues for future research.

II. RELATED WORK

Some related researches work on IoT or voice-controlled mechanisms implemented via different micro-controllers and systems. These systems have their own different approaches and building blocks for detailed research and descriptive analysis of using these specific technologies. They are similar regarding VIoT but in some areas, VIoT has been improved throughout and thus outweighs these previous researches.

This paper is situated within the broader landscape of robotic control and surveillance systems. A comparative analysis is drawn with existing literature, including a study on “A surveillance robocar employing IoT technology and the Blynk app” [1] for remote control and monitoring in challenging environments, particularly for military applications.

The research “IoT Based Voice Controlled Autonomous Robotic Vehicle Through Google Assistant” [2] shares similarities with the research endeavor, particularly in the realm of voice-controlled automation in robotic vehicles. Both researches aim to leverage the Internet of Things (IoT) to facilitate seamless communication between users and robotic vehicles, allowing for intuitive control mechanisms.

The research, “Voice Controlled Robotic Car Using Mobile Application” [3] shares common ground with VIoT by focusing on creating intuitive interfaces for users to interact with robotic vehicles, aiming to enhance accessibility and improve the quality of life for individuals, particularly those with physical disabilities.

The research, “Implementation of Human Voice Controlled Robotic Car” [4] aligns closely with the objectives outlined in the study of recognizing human voice instructions for controlling a robotic vehicle similar to VIoT. Both researches aim to develop systems capable of interpreting human voice commands to operate robotic vehicles, demonstrating the potential for intuitive human-robot interaction.

The research “Smart Buggy: An IoT-Based Smart Surveillance Robotic Car Using Raspberry Pi” [5] and “Design and Realization of Mobile Video Surveillance Car Based on ARM9 and Linux Platform” [6], focus on surveillance robotic cars controlled remotely via the internet.

“Automatic Monitoring and Controlling of Wi-Fi Based Robotic Car” [7] focuses on designing and implementing Wi-Fi-based robotic cars for remote monitoring and control. “IoT Driven Smart Car Integration with Google Firebase for Empowering Premises Security” [8] focuses on utilizing robotic cars equipped with various sensors for premises security and monitoring.

In “Automated Voice Controlled Car Using Arduino with Camera” [9], both researchers focus on voice-controlled cars with cameras, utilizing similar components. “An IoT Model for Autonomous Cars” [10] explores the integration of IoT technology into autonomous vehicle systems.

In “Real-Time Hand-Gesture Recognition for the Control of Wheelchair” [11] and Finger-Gesture Controlled Wheelchair with Enabling IoT [12], both researchers aim to enhance the accessibility and usability of mobility aids through innovative control systems.

III. SYSTEM METHODOLOGY

The methodology encompasses various stages, including hardware setup, software development, integration of voice recognition systems, and establishment of communication protocols. Through a systematic approach, we detail the steps taken to achieve this paper’s objectives, ensuring the robustness, efficiency, and functionality of the robotic system

The system architecture is structured into four distinct sectors to facilitate a comprehensive understanding of its functionality. Firstly, the software design section serves as the user interface that enables interaction and command input. Next, the Communication Systems handle data exchange between the app and the robotic car, employing IoT and Bluetooth technologies. The Hardware systems encompass the physical components of the robotic car, including microcontrollers, motors, and sensors. Lastly, the Issues and Solutions sector addresses challenges encountered during system operation and presents corresponding resolutions to ensure smooth functionality.

A. *Software Design*

The software section consists of an Android app having incorporates a singular button that, when pressed, initiates the establishment of connections with both the Firebase and Web Socket databases. This functionality is designed to save user data securely while serving as a protective measure against unauthorized commands for enhanced security. Essentially functioning as an ON-OFF switch, this streamlined approach ensures simplicity and effectiveness in managing connections and safeguarding the vehicle from undesired inputs.

The app enables users to view the video stream transmitted by the vehicle through the utilization of WebSocket technology. However, it's important to note that the video stream may experience high latency, primarily attributed to the substantial data size involved. Despite potential delays, users can access a live feed of the vehicle's perspective, enhancing the overall monitoring and interaction experience. Fig. 1. represents the system of the Google Speech Recognition API that actively captures and interprets user voice commands, converting spoken language into text strings.

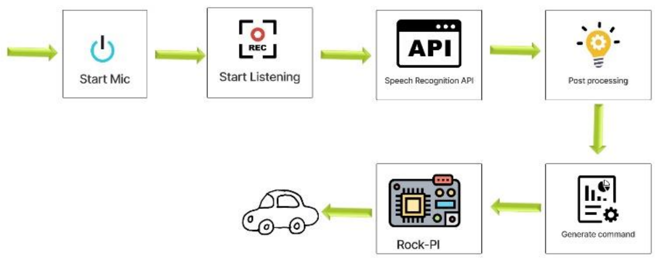

Fig. 1. Voice control process

To refine the accuracy of the commands, the Levenshtein algorithm, also known as the Edit Distance algorithm, comes into play. This dynamic programming technique measures the similarity between two strings, calculating the minimum number of single-character edits required to transform one string into another, thereby elevating the precision of recognizing spoken commands. The optimized strings undergo a conversion process, transforming them into integer values: Forward (3), Backward (4), Left (1), Right (2), and Stop (5). These integer values are then transmitted through both the WebSocket database for IoT communication and the Bluetooth system if within range. The microcontroller seamlessly retrieves and interprets these values, ensuring precise and effective execution of voice commands for controlling the robotic car.

Tailored for the Bengali language, the model encompasses five distinct commands as mentioned in Fig. 2.:

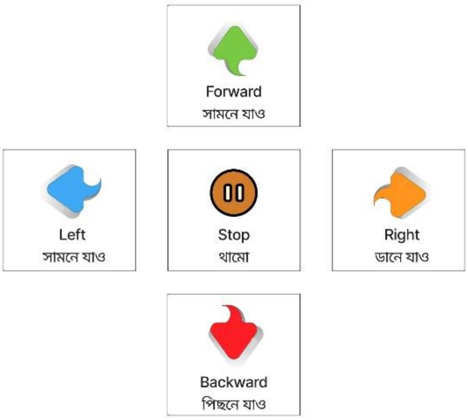

Fig. 2. Control and command mapping for remote control and Bengali voice control

Much like the Levenshtein-processed voice commands, the remote controller system employs a streamlined approach by converting button-pressed inputs into corresponding integer values: Forward (3), Backward (4), Left (1), Right (2), and Stop (5). These integer values are then transmitted through both the WebSocket database for IoT communication and the Bluetooth system if within range. Subsequently, the microcontroller reads and interprets these values, facilitating the effective execution of commands in the remote-control system.

The log cat feature in the system plays a pivotal role by displaying essential information regarding connection and communication status. It facilitates back communication initiated by the micro-controller, providing insights into the executed commands and promptly notifying any encountered errors. This comprehensive logging mechanism ensures transparency and real-time monitoring of system activities, contributing to a streamlined and error-aware operational environment.

B. Communication Systems

The system utilizes IoT (Internet of Things) and Bluetooth for communication. IoT enables remote control and data exchange over the internet, while Bluetooth allows for direct, short-range communication between the user's device and the car vehicle. These methods offer flexibility and accessibility in controlling the car in various environments. Fig. 3. represents the long-range data communication in the system that is achieved through internet access, employing WebSocket for efficient command transmission.

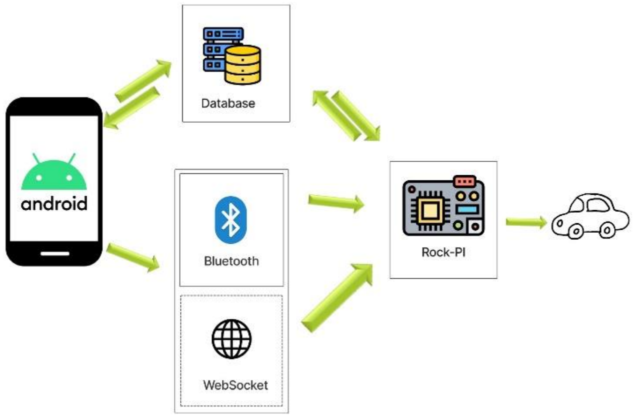

Fig. 3. IoT communication system

User inputs, whether through voice commands via the Android app or remote controller inputs, are processed and sent as command data through WebSocket. The microcontroller reads this data and executes corresponding actions, with the overall communication flow influenced by internet speed and read-write operations.

The latency of the system, affected by these factors, can be optimized with high-speed internet access in both the app and micro-controller, although the inherent latency is significantly lower when utilizing WebSocket compared to the Firebase database. This choice reflects a trade-off, prioritizing reduced latency over heightened data protection.

Fig. 4. demonstrates short-range data communication where the system employs Bluetooth connectivity, accessed locally with very low to no latency.

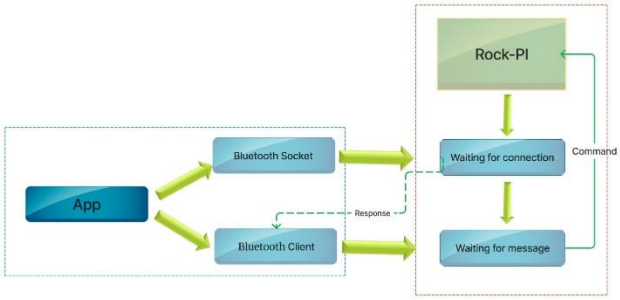

Fig. 4. Bluetooth communication system

The microcontroller, within an approximate 20-meter radius, actively accepts connections and continually checks for a local connection status. In the event of an error, the system attempts to reconnect every 2.5 seconds, ensuring a seamless and responsive short-range communication experience.

C. Hardware Design

This paper integrates four primary hardware components to facilitate its operations. At the core of its functionality lies the Rock Pi 3a Micro-controller, an advanced iteration of the Raspberry Pi, designed to serve as the central processing unit. Powered by the Rock Chip RK3328 processor featuring a quad-core Cortex-A55 ARM processor clocked at 2GHz, it boasts additional components such as a dedicated GPU and a wireless module. In total, the microcontroller offers 40 pins, including 28 GPIO pins, providing ample connectivity options for interfacing with various peripherals and sensors.

The L293D Motor Driver plays a pivotal role in bridging the microcontroller with the motors of the robotic vehicle. This component serves as the interface responsible for ensuring precise control over the motors' direction and speed. By translating GPIO signals from the microcontroller into actionable commands for the motors, the L293D Motor Driver facilitates seamless coordination between the hardware components, enabling smooth and responsive movement of the robotic vehicle.

In addition to the microcontroller and motor driver, this paper leverages a webcam to capture forward-facing footage. Employing a streaming method, the webcam feeds data to the microcontroller, which processes and transmits it via Web Socket. However, challenges related to latency may arise due to the transmission of large data volumes. Despite these potential hurdles, the webcam serves as a vital component for providing real-time visual feedback, enhancing the overall functionality and situational awareness of the robotic system.

D. Issues and solutions

This paper faces challenges related to the simultaneous arrival and departure of multiple types of data, potentially leading to conflicts in scheduling, increased latency, and the risk of losing overlapped data. To address these issues, a comprehensive solution has been implemented. Parallel execution is managed within the microcontroller through the deployment of two distinct scripts. The Camera Stream script runs independently to handle large camera data, minimizing latency impact on normal data exchange from Python scripts. Meanwhile, multi-threading is employed in the Python scripts, featuring four different threads: Write/Update, Bluetooth, Socket/IoT, and Read. The Write/Update thread manages responses to the database as log cats, the Bluetooth thread handles data initialization and exchange from Bluetooth, the Socket/IoT thread handles data related to IoT communication, and the Read thread, though not implemented, serves as a backup for exchanging extra data with the app. Notably, data transmission to the L293D motor driver via GPIO pins is executed serially with the threads, ensuring coordinated execution in scenarios involving multiple threads such as Bluetooth and IoT.

IV. EXPERIMENTAL ANALYSIS

In this section, the paper's improvements in various sectors are demonstrated in two different ways. Firstly, in terms of data passing with database and Web Socket and then the improvements compared to previous researches.

A. Comparison between using a database and a Web Socket

The data graph in Fig. 5. compares our implementation of WebSocket and using a normal database such as Firebase for IoT communication.

The dataset in Fig. 5. reveals the delays for both the WebSocket (Socket) and the Database (DB) in this paper. Calculating the averages, we find that the WebSocket demonstrates an average delay of approximately 499 ms, while the Database has an average delay of around 1534 ms. This suggests that, on average, the WebSocket exhibits a lower delay compared to the Database in the paper.

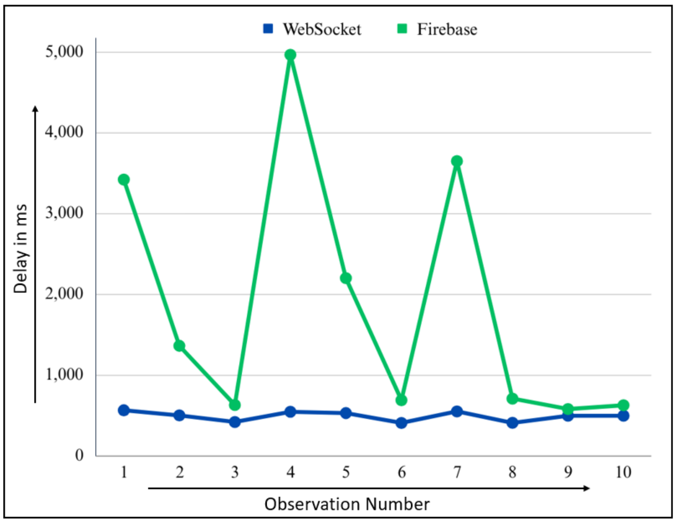

Fig. 5. Comparison between WebSocket and Firebase Delays in ms for 10 separate data exchanges

However, examining the individual entries, it becomes evident that there is inconsistency in the Database delay. For instance, Entry 4 shows a significantly higher DB delay of 4967 ms compared to the other entries. This inconsistency highlights the variability in delay times for database operations, indicating unpredictability in response times.

The advantages of utilizing WebSocket in this paper are multifold. Firstly, the WebSocket boasts a lower average delay, indicating faster and more responsive communication. Its real-time communication capabilities are especially crucial for applications like robotic control, where immediate responses to user commands are essential. Additionally, the WebSocket's reduced latency is beneficial for ensuring smooth and timely data exchange between the app and the microcontroller. This makes it an efficient choice for the IoT aspects of this paper, providing a lightweight and effective protocol for data transfer. Furthermore, the WebSocket delays appear more consistent across entries, suggesting a higher level of stability and predictability compared to the database.

Thus, the WebSocket integration in this paper stands out for its superior performance in terms of lower delays, real-time communication, and consistency. These factors make it a more favorable choice, particularly for time-sensitive applications such as controlling a UAV.

B. Comparison with previous researches

The research "Surveillance Robocar Using IoT and Blynk App" [1] demonstrates the feasibility of IoT-driven surveillance systems, VIoT extends this research by introducing the integration of voice commands alongside traditional remote controller inputs through the Android app. Furthermore, VIoT explores advanced features such as long-range IoT control capabilities and short-range Bluetooth interactions, underscoring its versatility and adaptability across diverse control scenarios.

VIoT extends beyond the scope of "IoT Based Voice Controlled Autonomous Robotic Vehicle Through Google Assistant" [2] integrating a broader range of voice commands, including commands in the Bengali language. Additionally, this paper enhances versatility by exploring features such as the integration of Bluetooth for short-range communication.

While the prototype “Voice Controlled Robotic Car Using Mobile Application” [3] utilizes voice recognition through Android phones and Bluetooth communication to control the robotic car's movements, this paper expands upon this foundation by integrating advanced features such as multilingual support and dual-mode control through both voice commands and a mobile application. Additionally, this paper explores the optimization of voice recognition algorithms, leveraging technologies like the Levenshtein algorithm for improved accuracy in interpreting user commands.

While “Implementation of Human Voice Controlled Robotic Car” [4] utilizes an Android app and UART protocol for transmitting voice instructions to the microcontroller, this paper expands upon this framework by incorporating advanced voice recognition algorithms and leveraging IoT technologies for seamless communication between the user interface and the robotic car. Additionally, this paper focuses on both long and short-range communications, utilizing effective solutions to enhance the accessibility and usability of voice-controlled robotic systems.

This paper stands out with advanced features compared to “Smart Buggy: An IoT Based Smart Surveillance Robotic Car Using Raspberry Pi” [5] and “Design and Realization of Mobile Video Surveillance Car Based on ARM9 and Linux Platform” [6] like voice and remote control, offering enhanced flexibility and usability compared to traditional surveillance trolleys.

This paper offers additional features like voice and IoT control, enhancing the functionality and versatility of the robotic car compared to a traditional Wi-Fi-based setup of “Automatic Monitoring and Controlling of Wi-Fi Based Robotic Car” [7]. Again, this paper stands out with additional layers of versatility and accessibility in controlling and monitoring the robotic car remotely, thereby enhancing overall functionality and user experience compared to “IoT-Driven Smart Car Integration with Google Firebase for Empowering Premises Security” [8].

This paper expands on the concept of “Automated Voice Controlled Car Using Arduino with Camera” [9] by integrating additional features such as long-range IoT control, short-range Bluetooth interactions, and support for Bengali voice commands, enhancing functionality and versatility for diverse control scenarios. While “An IoT Model for Autonomous Cars” [10] presents an IoT approach as a supplement to existing sensor sets, this paper offers a comprehensive system with advanced features like voice control and remote monitoring, enhancing functionality for more versatile applications.

While “Real-Time Hand-Gesture Recognition for the Control of Wheelchair” [11] and “Finger-Gesture Controlled Wheelchair with Enabling IoT” [12] focus on gesture recognition for smart wheelchairs, this paper incorporates advanced voice and remote-control functionalities, offering users greater flexibility and ease of use in navigating their environment.

Table I shows a shorter and summarized view of the mentioned reference researches and compares features with them to show the improvements and betterments in VIoT.

TABLE I
COMPARISON OF WORK IN IOT AND VOICE-CONTROLLED ROBOTIC VEHICLES

Reference No	**Voice Control**	**Bluetooth**	**Reduced Latency**	**IoT**
[1]	No	No	No	Yes
[2]	Yes	No	No	Yes
[3]	Yes	No	No	No
[4]	Yes	No	No	No
Proposed Method	Yes	Yes	Yes	Yes

Table I compares and summarizes all the different key features, focusing on aspects such as voice control, UART usage, Bluetooth connectivity, latency reduction, and IoT integration. References [1] to [4] represent existing approaches, with varying levels of support for these features. The proposed method, outlined in the last row, incorporates voice control, Bluetooth, and IoT capabilities, offering a comprehensive solution for robotic control. Notably, it reduces latency, further enhancing the system's functionality and performance. This comparison highlights the advancements and improvements introduced by the proposed method, positioning it as a promising solution in robotic communication and control.

V. CONCLUSIONS

This paper introduces a transformative approach to robotics, integrating Voice, IoT, and Bluetooth technologies to revolutionize human-robot interaction. Our system offers unparalleled flexibility and control, catering to diverse user preferences and needs. Beyond robotic control, the implications extend to assistive technology, where IoT and voice control can enhance accessibility for individuals with mobility impairments. For example, our system could be adapted to create an IoT-controlled wheelchair, enabling users to navigate their environment with simple verbal commands. Additionally, remote monitoring capabilities provide caregivers with real-time insights into user mobility patterns. This paper’s innovative approach holds promise for applications in smart homes, industrial automation, and healthcare robotics, driving inclusivity and empowering individuals to lead more independent lives.

This paper's future trajectory includes several enhancements to bolster its capabilities. Advanced voice recognition models will be integrated to improve accuracy and support multiple languages, enhancing accessibility. Autonomous navigation features like obstacle detection and collision avoidance will be developed to navigate diverse environments effectively. Exploring edge computing will reduce latency and enhance responsiveness, while additional accessibility features like gesture-based controls will promote inclusivity. In conclusion, this paper demonstrates its potential to pave the way for more advanced and inclusive robotic control interfaces, making significant strides in the realm of human-robot interaction.

REFERENCES

[1] Kumar, N., Dey, B., Chetri, C., Biswas, A. (2020). Surveillance Robocar Using IoT and Blynk App. In: Bera, R., Pradhan, P.C., Liu, CM., Dhar, S., Sur, S.N. (eds) Advances in Communication, Devices and Networking. ICCDN 2019. Lecture Notes in Electrical Engineering, vol 662. Springer, Singapore. https://doi.org/10.1007/978-981-15-4932-8_47

[2] M. Gupta, R. Kumar, R. K. Chaudhary and J. Kumari, "IoT Based Voice Controlled Autonomous Robotic Vehicle Through Google Assistant," *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2021, pp. 713-717, doi: 10.1109/ICAC3N53548.2021.9725526.

[3] 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.

[4] R. L. Khan, D. Priyanshu and F. S. Alsulaiman, "Implementation of Human Voice Controlled Robotic Car," *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*, MORADABAD, India, 2021, pp. 640- 646, doi: 10.1109/SMART52563.2021.9676319.

[5] S. M. Sali and K. R. Joy, "Smart Buggy: An IoT Based Smart Surveillance Robotic Car Using Raspberry Pi," *2021 8th International Conference on Smart Computing and Communications (ICSCC)*, Kochi, Kerala, India, 2021, pp. 280-285, doi: 10.1109/ICSCC51209.2021.9528144

[6] Y. Chai and J. Wang, "Design and Realization of Mobile Video Surveillance Car Based on ARM9 and Linux Platform," *2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC)*, Xi'an, China, 2018, pp. 1017-1021, doi: 10.1109/IMCEC.2018.8469452.

[7] S. Singh, M. Dilshad, M. Pushpavalli, P. Abirami, M. Kavitha and R. Harikrishnan, "Automatic Monitoring and Controlling of Wi-Fi Based Robotic Car," *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, Chennai, India, 2023, pp. 1-6, doi: 10.1109/ACCAI58221.2023.10199457.

[8] H. K. Abdulla, H. K. Omar, A. S. H. Abdul-Qawy and A. O. Ali, "IoT-Driven Smart Car Integration with Google Firebase for Empowering Premises Security," 2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 2023, pp. 387-394, doi: 10.1109/MIUCC58832.2023.10278358.

[9] C. Thirumarai Selvi, N. Anishviswa, G. A. Karthi, K. Darshan and M. G. Balaji, "Automated Voice Controlled Car Using Arduino with Camera," *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2021, pp. 708-711, doi: 10.1109/I-SMAC52330.2021.9640688.

[10] A. R. Boglut, F. Alexa, C. Caleanu and D. L. Lacrama, "An IoT Model for Autonomous Cars," *2023 31st Telecommunications Forum (TELFOR)*, Belgrade, Serbia, 2023, pp. 1-4, doi: 10.1109/TELFOR59449.2023.10372708.

[11] M. R. Huda, M. L. Ali and M. S. Sadi, "Real-Time Hand-Gesture Recognition for the Control of Wheelchair," *2022 12th International Conference on Electrical and Computer Engineering (ICECE)*, Dhaka, Bangladesh, 2022, pp. 384-387, doi: 10.1109/ICECE57408.2022.10088702.

[12] Sadi, M.S.; Alotaibi, M.; Islam, M.R.; Islam, M.S.; Alhmiedat, T.; Bassfar, Z. Finger-Gesture Controlled Wheelchair with Enabling IoT. Sensors 2022, 22, 8716. https://doi.org/10.3390/s22228716