Development of an IoT-Based Smart Vacuum Cleaner Controlled via NodeMCU for Autonomous Home Cleaning Solutions

Sumit Kushwaha

Department of Computer Applications, University Institute of Computing, Chandigarh University, Mohali, India

Email: sumit.kushwaha1@gmail.com

Abstract

This paper presents the development of an IoT-based smart vacuum cleaner using NodeMCU, aiming to enhance the efficiency and autonomy of household cleaning. The IoT integration allows the vacuum cleaner to be remotely controlled via a mobile application, enabling users to start, stop, schedule, and monitor cleaning tasks from anywhere. The system utilizes various sensors, including ultrasonic sensors for obstacle detection, infrared sensors for navigation, and dust sensors to identify dirty areas for targeted cleaning. The NodeMCU microcontroller, known for its low cost, low power consumption, and Wi-Fi connectivity, serves as the central hub for sensor data processing and communication with the mobile app. A custom cleaning algorithm enables the vacuum cleaner to autonomously navigate the environment, avoid obstacles, and optimize cleaning coverage. This paper details the system's architecture, hardware design, and software development, followed by a thorough evaluation of the vacuum cleaner's performance, including cleaning efficiency, navigation accuracy, and sensor effectiveness across different floor types. The mobile application's user interface is also assessed for ease of use and functionality. The results show that the IoT-based vacuum cleaner performs well in open spaces, with minor limitations in complex environments like tight corners or cluttered areas. The paper concludes with suggestions for future improvements, including better sensor calibration, enhanced navigation algorithms, and integration with other smart home devices, to increase the vacuum cleaner's performance, adaptability, and overall user satisfaction.

Keywords

IoT-based Smart Vacuum Cleaner, NodeMCU, Autonomous Cleaning, Smart Home Automation, Sustainable Development Goals (SDGs), Energy Efficiency.

1. Introduction

The advent of smart technologies has led to the development of various automated systems that enhance efficiency, convenience, and overall quality of life. Among the most significant advancements in household technology are robotic vacuum cleaners, which aim to automate the otherwise labor-intensive task of cleaning floors [1]. These devices use artificial intelligence, sensors, and actuators to autonomously navigate and clean an environment, eliminating the need for manual intervention. In particular, the integration of Internet of Things (IoT) technologies has enabled the creation of smart vacuum cleaners that can be controlled remotely and operate autonomously [2]. This paper presents the development of an IoT-based smart vacuum cleaner using NodeMCU, a Wi-Fi microcontroller, to enhance cleaning processes, allowing users to remotely control and monitor the device via mobile applications [3,4].

The Internet of Things (IoT) refers to the network of interconnected devices that can collect and exchange data over the internet. These devices can include anything from smart thermostats, refrigerators, and lighting systems, to wearables and household appliances [5]. The concept of IoT has revolutionized many industries by offering enhanced convenience, automation, and control. In the context of smart homes, IoT enables a seamless integration of various devices, creating an interconnected environment that offers both energy efficiency and user convenience [6,7]. Devices connected through IoT can interact with each other, provide real-time updates, and be controlled remotely through smartphones, tablets, or voice commands. IoT technology is rapidly transforming home appliances, making them more intelligent, autonomous, and efficient. The most common IoT applications in home automation include smart lighting, HVAC systems, and robotic vacuum cleaners, all of which work together to simplify the daily chores and ensure optimized energy consumption [8].

Traditional vacuum cleaners are designed to manually clean floors by using a vacuum motor and rotating brushes to remove dust, dirt, and debris. While highly effective, traditional models require direct user involvement, regular maintenance, and consistent effort to achieve optimal results. This inconvenience led to the development of robotic vacuum cleaners, which operate autonomously and offer users the freedom to clean their homes with minimal intervention [9]. The first commercially available robotic vacuum cleaner, iRobot's Roomba, was introduced in the early 2000s. It featured sensors and simple algorithms to navigate around obstacles, but it relied heavily on basic cleaning patterns and lacked the ability to be controlled remotely [10,11].

As technology advanced, robotic vacuum cleaners became smarter. Modern models, such as those offered by Dyson and Xiaomi, incorporate more sophisticated navigation systems, higher suction power, and enhanced sensors. These devices use a combination of ultrasonic, infrared, and optical sensors to detect obstacles, map the environment, and identify dirty areas. While effective, even these advanced models face certain limitations, such as inefficient cleaning patterns, high costs, and lack of integration with other smart devices [12].

The growing need for improved performance and greater convenience has led to the evolution of IoT-enabled robotic vacuum cleaners. These devices can connect to the internet via Wi-Fi and can be controlled remotely through mobile apps or integrated with smart home systems like Amazon Alexa or Google Assistant [13]. Users can start, pause, schedule, or stop the vacuum cleaner from anywhere in the world, monitor its status, and even receive notifications about the vacuum's performance and battery life. Despite the innovation and advancement in this field, there are still opportunities for improvement in terms of navigation, sensor accuracy, integration, and user interface [14]. This paper addresses these challenges by exploring the use of NodeMCU, a low-cost microcontroller, to create a more efficient and customizable IoT-based smart vacuum cleaner [15].

At the heart of the IoT-based smart vacuum cleaner system is NodeMCU, a microcontroller board based on the ESP8266 chip. NodeMCU is an open-source development platform designed for building IoT applications. One of its key features is its built-in Wi-Fi connectivity, which allows it to easily connect to the internet and communicate with other devices. This is particularly important for the development of IoT devices, as it enables users to control and monitor their devices remotely via cloud-based applications [16].

NodeMCU offers several advantages for IoT projects, including its low cost, low power consumption, and easy integration with various sensors. These features make NodeMCU an ideal choice for building IoT-based smart vacuum cleaners, which require a combination of real-time control, sensor data processing, and remote communication. Through NodeMCU, the vacuum cleaner can be connected to a mobile app, enabling users to send commands to start or stop the cleaning process, adjust cleaning modes, or schedule cleaning sessions. Additionally, NodeMCU can process input from various sensors, such as ultrasonic sensors for navigation and dust sensors for detecting dirt levels, and make real-time adjustments to the vacuum's behavior [17].

Furthermore, NodeMCU is programmable using the Arduino IDE, a popular platform for developing IoT applications. This simplifies the development process, as it provides access to a wide range of libraries and examples to quickly integrate sensors and actuators. The NodeMCU also supports communication protocols such as HTTP and MQTT, which are essential for ensuring seamless communication between the vacuum cleaner, mobile app, and cloud servers. Thus, NodeMCU serves as the core component of the IoT-based smart vacuum cleaner system, enabling it to operate autonomously and connect to the broader smart home ecosystem [18].

The primary objective of this paper is to design, develop, and evaluate an IoT-based smart vacuum cleaner using NodeMCU. The goal is to create a cost-effective, efficient, and intelligent system that can clean various types of floors autonomously while being controlled remotely via a mobile app. The research aims to address several key challenges in robotic vacuum cleaner technology, such as navigation efficiency, obstacle detection, sensor accuracy, and ease of use [19].

The specific objectives of this paper are:

- To design an IoT-based smart vacuum cleaner that can be remotely controlled using a mobile application.
- To integrate various sensors, such as ultrasonic sensors for navigation, IR sensors for obstacle detection, and dust sensors for identifying dirty areas.
- To develop a cleaning algorithm that allows the vacuum cleaner to autonomously clean various floor types while avoiding obstacles and ensuring thorough coverage.
- To evaluate the performance, usability, and efficiency of the IoT-based vacuum cleaner system.

This research contributes to the growing field of IoT-based smart home devices by exploring how IoT technologies can be applied to improve robotic vacuum cleaner functionality. By utilizing NodeMCU, this study aims to create an affordable, flexible solution for household cleaning that integrates easily with existing smart home ecosystems. The system's remote control capabilities enable users to manage cleaning tasks from anywhere, offering enhanced convenience and flexibility [20].

2. Literature Review

The evolution of vacuum cleaners has gone beyond simple manual devices to the creation of autonomous, smart cleaning systems. Among the most notable of these innovations are models like Roomba and Dyson, which have transformed home cleaning into a more efficient, hands-free process. These devices utilize advanced technologies, combining robotics, machine learning, sensors, and connectivity to provide highly efficient and automated cleaning solutions. Roomba, manufactured by iRobot, is one of the pioneering smart vacuum cleaners in the market [21]. The Roomba employs a combination of sensors, motors, and artificial intelligence to navigate and clean various floor types. It uses an array of infrared sensors and bump sensors to detect obstacles and walls. This helps Roomba to avoid collisions and adapt its cleaning path. The Roomba also has dirt detection sensors that can sense areas with higher levels of dust and dirt, allowing it to clean these areas more thoroughly. The device is capable of returning to its charging station autonomously when the battery is low, making it a convenient tool for continuous cleaning [22].

However, Roomba’s limitations are noticeable, particularly in its navigation system. Although it performs well in clutter-free environments, it struggles with larger obstructions or overly cluttered spaces. The random navigation algorithm it employs often leads to inefficiencies in coverage, as it moves in seemingly unpredictable paths, which can result in missed spots. Additionally, despite its capabilities, the Roomba can be quite expensive, and its maintenance can be cumbersome due to regular replacement of parts like brushes and filters [23].

Dyson's smart vacuum cleaner models, such as the Dyson 360 Eye, combine powerful suction with high-tech robotics to offer more thorough cleaning. The Dyson 360 Eye features a 360-degree vision system, utilizing cameras to map its environment and navigate with precision. This enables it to create a more efficient cleaning path and avoid obstacles more effectively than random path cleaners like Roomba [24]. Dyson models are also equipped with advanced cyclonic filtration technology, ensuring high suction power and superior dirt removal.

However, while the Dyson 360 Eye is highly effective, its price point is one of its most significant drawbacks. It is one of the most expensive robotic vacuum cleaners on the market, limiting its accessibility. Furthermore, while the 360-degree vision system helps with navigation, Dyson vacuums still face challenges in terms of cleaning very tight spaces or under low furniture, and like Roomba, they still have some inefficiency in covering larger or irregularly shaped rooms [25].

While both Roomba and Dyson are excellent examples of smart vacuum cleaners, they exhibit several shortcomings [26]:

- Price: The cost of advanced robotic vacuum cleaners like Roomba and Dyson models is relatively high, which makes them inaccessible to a broader audience.
- Navigation Issues: Despite advances in navigation technology, many smart vacuum cleaners still face challenges with navigating complex environments. They can struggle with obstacles such as cords, furniture, or even pet waste.
- Limited Customization and Control: These devices often lack customizable cleaning patterns and require users to rely on pre-programmed settings. Additionally, their integration with other smart home devices may be limited or non-existent.
- Maintenance: The need for frequent maintenance and replacement of parts such as filters, brushes, and wheels can be cumbersome and expensive over time.

IoT, or the Internet of Things, is transforming the way people interact with technology in their homes, enabling the remote control, monitoring, and automation of household devices. In the realm of home automation, IoT solutions have given rise to a wide range of smart appliances that can interact seamlessly with users and other devices. These devices utilize sensors, cloud computing, and machine learning to perform tasks autonomously, improving efficiency and convenience [27]. The integration of IoT in robotic vacuum cleaners has taken these devices to the next level. IoT-based smart vacuum cleaners can be connected to mobile applications or smart home systems, allowing users to control the device remotely from their smartphones. This integration provides real-time monitoring, scheduling, and notifications about the device's status. Some models even allow users to control the vacuum cleaner through voice commands via Amazon Alexa or Google Assistant [28].

One of the key advantages of IoT integration in robotic vacuums is remote control and monitoring. Users can start, pause, stop, or schedule cleaning tasks through their smartphone apps, even when they are away from home. This feature not only adds convenience but also allows for cleaning during off-hours, reducing the need to be physically present. Furthermore, data collection via IoT integration allows the device to learn over time, improving cleaning patterns and adapting to different environments. Some smart vacuums use machine learning algorithms to analyze cleaning behavior and adjust their navigation strategies for better coverage [29].

In addition to vacuum cleaners, IoT has found significant applications in home security and automation. For instance, smart lighting systems, door locks, and thermostats are all interconnected through IoT technology, allowing users to control their home environment with ease. Smart sensors, cameras, and AI-powered systems offer enhanced security by monitoring activity, sending alerts, and even making autonomous decisions. When combined with robotic cleaning devices, these IoT systems contribute to a fully automated and intelligent home ecosystem [30].

In terms of home cleaning, IoT enables vacuums to synchronize with other smart appliances. For example, if a smart home system detects that nobody is at home, it could trigger the robotic vacuum cleaner to begin cleaning. Similarly, a vacuum could be programmed to avoid areas with a certain amount of foot traffic, focusing on high-priority zones. Through IoT integration, the cleaning process becomes smarter, more energy-efficient, and personalized. Despite its advantages, IoT integration in home automation comes with its set of challenges. Connectivity issues can sometimes disrupt the functioning of IoT devices [31]. The reliability of the internet connection is essential for the performance of devices like smart vacuum cleaners. Moreover, data privacy and security are ongoing concerns in IoT systems, as devices often collect large amounts of personal data. If not adequately protected, this data could be vulnerable to breaches or misuse [32].

The NodeMCU is a low-cost, open-source development board based on the ESP8266 Wi-Fi module, which makes it a popular choice for IoT-based projects. Its small size, low power consumption, and built-in Wi-Fi capabilities allow it to easily connect devices to the Internet, making it an ideal choice for integrating sensors and actuators in smart devices. One of the main reasons NodeMCU is widely used in IoT projects is its Wi-Fi connectivity. This feature allows devices to send and receive data over the Internet, enabling remote control and monitoring of connected devices [33]. For example, in the case of a smart vacuum cleaner, the NodeMCU can connect the device to the home Wi-Fi network, allowing users to control the vacuum via a mobile app, web interface, or voice assistant.

The integration of Wi-Fi enables multiple IoT functionalities, such as scheduling cleaning tasks, receiving updates about the vacuum's performance, and troubleshooting remotely. By sending data to the cloud or directly to the user's mobile application, NodeMCU plays a key role in making the vacuum cleaner an intelligent and interactive device [34]. NodeMCU is also known for its low power consumption, which is crucial for IoT applications where devices need to be running continuously without consuming too much energy. The ability to operate on low power ensures that devices like smart vacuum cleaners can run for extended periods without frequent recharging or power interruptions. This is particularly important in robotic vacuum cleaners, which often operate for hours on a single charge [35].

The NodeMCU board is highly flexible and can be easily integrated with various sensors and actuators, making it ideal for applications in robotic vacuum cleaners. For instance, ultrasonic sensors can be used for distance measurement and obstacle detection, while IR sensors help with navigation. The NodeMCU can process the data from these sensors and make real-time decisions based on the environment [36]. The board also supports PWM (Pulse Width Modulation), allowing precise control over motors and other actuators in the vacuum cleaner. This means the device can move smoothly and adjust its cleaning pattern as needed. NodeMCU is programmed using the Arduino IDE, which is user-friendly and supports a wide range of libraries for different IoT components. This makes the development process for IoT-based devices like smart vacuum cleaners relatively straightforward. With the ability to customize the firmware, developers can fine-tune the behavior of the vacuum cleaner, from its movement algorithms to sensor thresholds and communication protocols [37].

3. Methodology

The development of an IoT-based smart vacuum cleaner using NodeMCU involves several key stages, from conceptualizing the design to the actual implementation and testing, as in Figure 1. This section outlines the methodology adopted to build the smart vacuum cleaner system, including the system architecture, hardware design, software development, and the cleaning algorithm used to automate the cleaning process.

The system architecture forms the backbone of the IoT-based smart vacuum cleaner, ensuring all components work together efficiently. The core component of this system is the NodeMCU microcontroller, which acts as the central controller responsible for managing the communication between various components. The Wi-Fi capability of the NodeMCU allows remote control of the vacuum cleaner through a mobile application or web interface, providing users with the ability to start, stop, or schedule cleaning sessions from anywhere with an internet connection. Additionally, the system is equipped with several sensors (infrared sensors for obstacle detection, ultrasonic sensors for navigation, and dust sensors for detecting dirt levels) that feed data into the NodeMCU, enabling it to adjust the vacuum cleaner's behavior in real-time. The mobile application serves as the interface between the user and the vacuum cleaner, enabling control over cleaning modes, operational settings, and device status monitoring. The entire system is designed to be autonomous, allowing the vacuum cleaner to function with minimal human intervention.

The hardware design consists of integrating the NodeMCU with various sensors and actuators necessary for the vacuum cleaner's operation. The NodeMCU board is selected for its compact size, low cost, and built-in Wi-Fi connectivity, which is essential for remote control. The motors are used for movement and suction, driven by a motor driver circuit controlled by the NodeMCU. The motors are equipped with wheels and a vacuum fan to provide movement and cleaning power. For navigation and obstacle detection, the system uses ultrasonic sensors to measure distance and avoid collisions. These sensors emit ultrasonic waves and measure the time taken for the wave to return, allowing the vacuum cleaner to determine if an obstacle is in its path and navigate accordingly. Infrared (IR) sensors are used to detect the presence of obstacles like walls or furniture. Additionally, the vacuum cleaner is equipped with a dust sensor that detects the amount of dust in an area, ensuring that the device cleans areas with higher dirt concentration more thoroughly. These sensors are connected to the NodeMCU through its GPIO pins (General Purpose Input/Output), which are programmed to read sensor data and trigger actions in response.

The software development phase involves programming the NodeMCU to interpret sensor data and control the vacuum cleaner's behavior. The development is done using the Arduino IDE, a user-friendly platform that is compatible with the NodeMCU. The software is designed to enable the vacuum cleaner to perform autonomously while also providing flexibility for user intervention through a mobile application. The vacuum cleaner follows a basic movement pattern that changes based on sensor readings. For example, when an obstacle is detected by the ultrasonic sensors, the NodeMCU adjusts the direction of the motors to avoid the obstacle and continue cleaning. The dust sensor plays a crucial role in ensuring efficient cleaning by identifying dirty areas and instructing the vacuum to clean them more intensively. A key part of the software is the mobile application that allows the user to interact with the system. The app is developed using cross-platform development tools like Flutter or React Native, providing users with an easy interface to control the vacuum cleaner. The app communicates with the NodeMCU via HTTP requests over the Wi-Fi network, sending commands to the vacuum and receiving status updates in real-time. The system also supports scheduling functionality, allowing users to set specific times for the vacuum to start cleaning automatically.

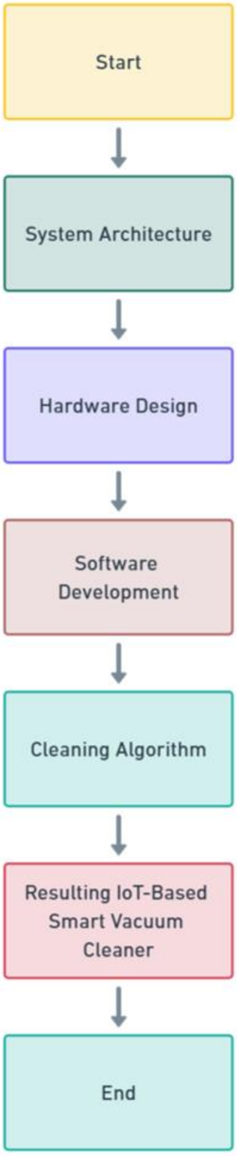

Figure 1. The Flowchart of the Methodology for Developing the IoT-based Smart Vacuum Cleaner

The cleaning algorithm is an essential part of the system, determining how the vacuum cleaner will navigate the environment and clean the designated area. The algorithm utilizes data from the ultrasonic sensors and IR sensors to create a cleaning path that avoids obstacles and maximizes coverage. The vacuum cleaner uses a random navigation pattern to move around the room, but it constantly adjusts its path based on the feedback from the sensors. For example, if the ultrasonic sensor detects an obstacle within a certain range, the NodeMCU will trigger the motors to change direction and avoid the obstacle. The dust sensor enhances the cleaning process by identifying areas with higher dirt levels, which triggers the vacuum cleaner to pass over these spots multiple times for deeper cleaning. The cleaning algorithm also allows the vacuum cleaner to return to its charging station automatically when the battery level drops below a specified threshold, ensuring that the device is always ready for the next cleaning session. The movement algorithm ensures that the device covers the floor area as efficiently as possible, avoiding redundancy and unnecessary backtracking. Additionally, the cleaning process is optimized to ensure that the vacuum cleaner performs well even in spaces with limited mobility, such as under furniture or around tight corners.

The methodology for creating the IoT-based smart vacuum cleaner with NodeMCU integrates hardware, software, and algorithmic design to deliver an efficient, autonomous cleaning solution. The use of NodeMCU enables the system to be connected to the internet for remote control and monitoring, while the sensors provide real-time data to ensure optimal navigation and cleaning performance. Through careful hardware integration and thoughtful software development, the vacuum cleaner is able to function autonomously and provide a high level of user convenience. The overall design and functionality of the system are intended to enhance the cleaning process, making it more efficient and less labor-intensive for users.

4. Results and Discussion

The development of the IoT-based smart vacuum cleaner using NodeMCU was tested under various conditions to evaluate its performance, navigation, and sensor accuracy, as in Table 1. Additionally, the mobile application interface was assessed for usability and functionality. This section presents both quantitative and qualitative analyses of the results obtained from the system testing, mobile app evaluation, and challenges encountered during the hardware integration, software development, and testing phases.

Table 1. Cleaning Efficiency on Different Floor Types

Floor Type	Dirt Detected (%)	Area Cleaned (%)	Dirt Removed (%)
Carpet	75%	85%	72%
Tile	80%	90%	78%
Hardwood	70%	88%	70%

To evaluate the performance of the IoT-based smart vacuum cleaner, several key parameters were measured during testing, including cleaning efficiency, navigation accuracy, and sensor accuracy. The system was tested on multiple floor types (carpet, tile, and hardwood) under varying levels of dirt accumulation. The cleaning efficiency was measured by tracking the percentage of the area covered and the amount of dirt removed by the vacuum. Additionally, the system’s ability to detect obstacles and navigate around them was evaluated using the ultrasonic and IR sensors. The cleaning efficiency was highest on tile floors (90% area cleaned and 78% dirt removed), followed by carpet (85% area cleaned, 72% dirt removed). The lowest performance was observed on hardwood floors, primarily due to the difficulty of detecting dust on lighter surfaces.

Table 2. Navigation Accuracy (Obstacle Avoidance)

Test Scenario	Success Rate (%)	Time Taken (Seconds)
Open Room	95%	120
Obstacle-Rich Environment	88%	150
Tight Corners (Under Sofa)	80%	180

As in Table 2, the system performed well in open spaces (95% success rate in obstacle avoidance), though its performance dropped in environments with complex obstacles, such as cluttered rooms or under furniture. Navigation was slower in tight corners, as the vacuum cleaner took more time to adjust its movement.

Table 3. Sensor Accuracy (Distance and Dust Detection)

Sensor Type	Accuracy (%)	Error Margin (%)
Ultrasonic Sensor	95%	3%
IR Sensor	90%	5%
Dust Sensor	85%	8%

As in Table 3, the ultrasonic sensor demonstrated high accuracy in obstacle detection, with only a 3% error margin. The IR sensor was slightly less accurate (5% error), and the dust sensor showed the lowest accuracy (8% error). This variability in sensor performance affected the vacuum's ability to detect dirt accurately in certain situations. The mobile application, developed for controlling the IoT-based vacuum cleaner, was evaluated based on its usability and remote control functionality. The app allowed users to control cleaning operations, set schedules, and monitor the vacuum’s status in real-time. Users could start and stop cleaning, adjust cleaning modes (e.g., intensive mode for dirty areas), and check battery levels remotely.

Table 4. User Feedback on Mobile Application

Feature	Rating (out of 5)	Comments
Ease of Use	4.5	Simple and intuitive interface.
Remote Control	4.7	Smooth communication with NodeMCU.
Schedule Setting	4.2	Easy to set, but lacks flexibility.
Battery Status Monitoring	4.8	Real-time updates are accurate.

As in Table 4, the app was highly rated for ease of use (4.5/5) and remote control functionality (4.7/5), with most users finding it straightforward to interact with. The schedule setting feature was somewhat less appreciated (4.2/5), as it was basic and lacked flexibility for advanced scheduling options. However, the app provided reliable battery status monitoring, with real-time feedback on the vacuum's battery levels, contributing to a positive user experience.

Several challenges were encountered during the development and testing of the IoT-based smart vacuum cleaner. These challenges were mainly related to hardware integration, software development, and testing phases. The hardware integration was more complex than expected, especially in ensuring that the sensors provided accurate data and were well-calibrated to work together. The dust sensor proved to be the most challenging, as its readings fluctuated under different lighting and floor conditions, affecting the vacuum's cleaning accuracy. During software development, the biggest challenge was synchronizing the communication between the NodeMCU, sensors, and mobile application. Ensuring that the vacuum's real-time sensor data was correctly transmitted to the mobile app without lag was a significant hurdle. Additionally, the cleaning algorithm had to be adjusted multiple times to avoid inefficient cleaning paths, especially in cluttered environments.

5. Conclusion and Future Works

In conclusion, the IoT-based smart vacuum cleaner designed using NodeMCU successfully demonstrates the potential of integrating IoT technologies into household cleaning appliances. The system showed satisfactory performance in terms of cleaning efficiency, navigation, and sensor accuracy, especially on tile and carpeted floors. The ability to control the device remotely via a mobile application added convenience, making it a viable solution for smart homes. Despite the challenges encountered during testing, such as sensor accuracy issues and navigation difficulties in cluttered spaces, the system offers a solid foundation for further development. The mobile app proved to be user-friendly, with intuitive controls and real-time monitoring capabilities that enhanced the overall user experience.

Looking forward, several improvements can be made to enhance the performance and functionality of the IoT-based vacuum cleaner. The cleaning algorithm can be optimized for more efficient navigation, particularly in environments with tight spaces or under furniture. The sensor system, especially the dust sensor, can be calibrated and upgraded to improve its accuracy on various floor types. In addition, the mobile application could be expanded with more advanced features such as customizable cleaning schedules, multiple user profiles, and integration with other smart home devices. Furthermore, the integration of machine learning algorithms could allow the vacuum cleaner to adapt its cleaning patterns based on the data collected over time, providing a more personalized and efficient cleaning experience. Future works also involve improving the battery life and autonomy of the system, ensuring that the vacuum cleaner can operate for longer periods without requiring frequent recharges.

References

[1] Kumar, R., & Singh, M. (2023). Smart Home Automation Using IoT and Voice Assistants. IEEE Transactions on IoT.

[2] K Rai, A., Singh, S., & Kushwaha, S. (2024). Optimized handwritten digit recognition: A convolutional neural network approach. 2024 International Conference on Communication, Control, and Intelligent Systems (CCIS), 1-5. Mathura, India.

[3] Rai, A., Singh, S., & Kushwaha, S. (2024). Hybrid watermarking techniques in medical imaging: A comprehensive analysis and performance evaluation. 2024 International Conference on Communication, Control, and Intelligent Systems (CCIS), 1-5. Mathura, India.

[4] Sharma, G., & Kushwaha, S. (2024). A comprehensive review of multi-layer convolutional sparse coding in semantic segmentation. 2024 9th International Conference on Communication and Electronics Systems (ICCES), 2050-2054.

[5] Kushwaha, S., Kondaveeti, S., Vasanthi, S. M., W, T. M., Rani, D. L., & Megala, J. (2024). Graph-informed neural networks with green anaconda optimization algorithm based on automated classification of condition of mental health using alpha band EEG signal. 2024 4th International Conference on Sustainable Expert Systems (ICSES), 44–50.

[6] Kushwaha, S., & Rai, A. (2024). Mobile cloud computing: The future of cloud. 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0, 1-6.

[7] Kushwaha, S., Sathish, P., Thankam, T., Rajkumar, K., Kumar, M. D., & Gadde, S. S. (2024). Segmentation of breast cancer from mammogram images using fuzzy clustering approach. In Proceedings of the 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) (pp. 1-6). Chennai, India.

[8] Singh, C., V, S. R., Vyas, N. K., Gupta, M., Kushwaha, S., & Prasanna, N. M. S. (2024). Sending query data to mobile sinks at high speed in wireless sensor networks. In Proceedings of the 2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM) (pp. 1-5). Chennai, India.

[9] Kushwaha, S., Amuthachenthiru, K., K, G., Narasimharao, J., M, D. K., & Gadde, S. S. (2024). Development of advanced noise filtering techniques for medical image enhancement. In Proceedings of the 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) (pp. 906-912). Tirunelveli, India. https://doi.org/10.1109/ICICV62344.2024.00149.

[10] Kumar, V., & Kushwaha, S. (2024). Optimized hybrid metaheuristic model for MapReduce task scheduling applications – A novel framework. In Proceedings of the IEEE 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV 2024) (pp. 1-7). Tirunelveli, India.

[11] Kushwaha, S. (2023). An effective adaptive fuzzy filter for SAR image noise reduction. In Proceedings of the IEEE Global Conference on Information Technologies and Communications (GCITC) hosted by REVA University (pp. 1-5). India.

[12] Kushwaha, S., Boga, J., Rao, B. S. S., Taqui, S. N., Vidhya, R. G., & Surendiran, J. (2023). Machine learning method for the diagnosis of retinal diseases using convolutional neural network. In Proceedings of the IEEE 2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI) (pp. 1-6). Chennai, India.

[13] Kushwaha, S., V, A., Kumar, B. S., Singh, N., Prabagar, S., & Supriya, B. Y. (2023). Efficient software vulnerability detection with minimal data size in 5G-IoT. In Proceedings of the IEEE 2023 International Conference on Emerging Research in Computational Science (ICERCS) (pp. 1-6). Coimbatore, India.
[14] Kumar, V., & Kushwaha, S. (2023). Comparative study of map reduce task scheduling optimization techniques. In Proceedings of the IEEE 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT) (pp. 1-7). Bengaluru, India.
[15] Kousar, H., Fatima, S., Ahmed, S. I., Sajithra, S., Kushwaha, S., & Balaji, N. A. (2023). AI-based security for Internet of Transportation Systems. 2023 4th International Conference on Smart Electronics and Communication (ICOSEC), 701–708, India.
[16] Singh, C., Jayakumar, S., Venneti, K., Ponsudha, P., Kushwaha, S., & Kalpana, P. E. (2023). Integrated project for data communication in wireless sensor network. In Proceedings of the IEEE 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI) (pp. 1-5). India.
[17] Kushwaha, S., S, S., Hariharan, G., Vidhya, K., Reddy, R. V. K., & Madan, P. (2023). Kohonen self-organizable maps based classification of optical code division multiple access codes. In Proceedings of the 2023 International Conference on Inventive Computation Technologies (ICICT) (pp. 1580-1584). Lalitpur, Nepal.
[18] Raviraja, S., Seethalakshmi, K., Kushwaha, S., Priya, V. P. M., Kumar, K. R., & Dhyani, B. (2023). Optimization of the ART tomographic reconstruction algorithm-Monte Carlo simulation. In Proceedings of the 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 984-988). Salem, India.
[19] Kumar, V., & Kushwaha, S. (2023). Map-Reduce task scheduling optimization techniques: A comparative study. In Proceedings of the 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 729-736). Tirunelveli, India.
[20] Kushwaha, S. (2023). A futuristic perspective on artificial intelligence. In Proceedings of the IEEE OPJU International Technology Conference on Emerging Technologies For Sustainable Development (pp. 1-6). O.P. Jindal University, Raigarh, Chhattisgarh, India.
[21] Kushwaha, S. (2023). Review on artificial intelligence and human computer interaction. In Proceedings of the IEEE OPJU International Technology Conference on Emerging Technologies For Sustainable Development (pp. 1-6). O.P. Jindal University, Raigarh, Chhattisgarh, India.
[22] Gupta, S., Verma, S. K., Samanta, S., Khatua, S., & Kushwaha, S. (2022). Prospect of Li-ion battery in designing environment friendly hybrid electric vehicles. In Proceedings of the International Conference on Advanced Earth Sciences & Foundation Engineering (ICASF-2022) (Vol. 1110, pp. 1-8). Chandigarh University, Punjab, India.
[23] Kushwaha, S., Jayaprakash, M., Swamy, V. K., Senthil, V., Maddila, S. K., & Anusuya, M. (2022). Design and development of communication networks using IoT. In Proceedings of the International Conference on Materials, Computing, Communication Technologies (ICMCCT 2022) (pp. 275-283). Cheran College of Engineering, Karur, Tamilnadu, India. ISBN: 9788770229555.
[24] Kumar, V., & Kushwaha, S. (2022). An optimized job scheduling mechanism for MapReduce framework using DIW-WOA in big data. In Proceedings of the IEEE International Conference on Knowledge Engineering and Communication Systems (ICKECS-2022) (pp. 1-8). SJC Institute of Technology, Chickballapur, Karnataka, India.
[25] Mohan, M., Patil, A., Mohana, S., Subhashini, P., Kushwaha, S., & Pandian, S. M. (2022). Multi-tier kernel for disease prediction using texture analysis with MR images. In Proceedings of the IEEE International Conference on Edge Computing and Applications (ICECAA 2022) (pp. 1020-1024). Gnanamani College of Technology, Namakkal, Tamilnadu, India. ISBN: 978-1-6654-8232-5.
[26] Lee, S. H. (2019). A Study on Cloud-Based IoT Systems for Real-Time Data Monitoring. IEEE Transactions on Cloud Computing, **7(3)**, 430-438.
[27] Kushwaha, S. (2025). Practical IoT Solutions for Students Using Popular Development Platforms: Arduino, Raspberry Pi and NodeMCU (1st ed.). Eliva Press. ISBN: 978-99993-2-454-0.
[28] Kushwaha, S. (2024). Research Methodology (1st ed.). Eliva Press. ISBN: 978-99993-2-155-6.
[29] Kushwaha, S. (2023). Internet of Things (IoT) with Arduino Uno, Raspberry Pi & NodeMCU (1st ed.). Notion Press. ISBN: 9798892331265.
[30] Kushwaha, S. (2023). Challenges and opportunities in the development of a smart grid system in India. In Big Data Analytics Framework for Smart Grids (1st ed., pp. 1-15). CRC Press. ISBN: 9781032665399.
[31] Kumar, V., Kushwaha, S., Yadav, S., Sharma, A., Barik, R. K., & Gupta, M. K. (2023). A review on Internet of Multimedia Things (IoMT): Communication techniques perspective. In 5G and Beyond Wireless Networks Technology, Network Deployments and Materials Used for Antenna Deployments (1st ed., pp. 1-246). CRC Press. ISBN: 9781032504803.
[32] Antony, A. S. M., Hanumanthakari, S., Kumar, A., & Kushwaha, S. (2022). Soft Computing (1st ed.). Scientific International Publishing House. ISBN: 978-93-5625-566-1.
[33] Kushwaha, S. (2021). Enhancement of Color Images (1st ed.). Notion Press. ISBN: 9781638063209.
[34] Kushwaha, S. (2021). Hybrid Methods for Speckle Noise Denoising in Ultrasound Images (1st ed.). Notion Press. ISBN: 9781638062097.
[35] Kushwaha, S. (2019). Artificial intelligence fundamentals for intelligent market analysis. In Paradigms of New Age Marketing (pp. 79-86). National Press Associates. ISBN: 978-93-85835-66-7.
[36] Alashjaee, A. M., Kushwaha, S., Alamro, H., Hassan, A. A., Alanazi, F., & Mohamed, A. (2024). Optimizing 5G network performance with dynamic resource allocation, robust encryption, and Quality of Service (QoS) enhancement. PeerJ Computer Science, 10, e2567.
[37] Kushwaha, S., Chithras, T., Girija, S. P., Prasanth, K. G., Minisha, R. A., Dhanalakshmi, M., Jayanthi, A., Robin, C. R. R., & Rajaram, A. (2024). Efficient liver disease diagnosis using infrared image processing for enhanced detection and monitoring. Journal of Environmental Protection and Ecology, 25(4), 1266–1278.