ABUAD Journal of Engineering Research and Development (AJERD)
ISSN (online): 2645-2685; ISSN (print): 2756-6811

Volume 8, Issue 3, 293-300

Development of AI Based Gesture and Voice Control Home Automation System

Nnamdi OKOMBA, AdedayoSOBOWALE, Adebimpe ESAN, Bolaji OMODUNNBI, Taiwo AWOYEMI, Oluwaseyi OGUNDIGBA

Department of Computer Engineering, Federal University Oye-Ekiti, Ekiti, Nigeria

nnamdi.okomba@fuoye.edu.ng/sobowaleadedayo@gmail.com/adebimpe.esan@fuoye.edu.ng/ bolaji.omodunbi@fuoye.edu.ng/awoyemitaiwoakinol@gmail.com/belovedoluwaseyi2000@gmail.com

Corresponding Author: nnamdi.okomba@fuoye.edu.ng, +2348037755791
Received: 28/09/2025
Revised: 23/11/2025
Accepted: 01/12/2025
Available online: 04/12/2025

***Abstract**: This study introduces the development and application of an AI-driven smart home system that combines voice and gesture controls to improve user convenience, accessibility, and energy conservation. The research aimed to create a system for managing home appliances through voice instructions and hand movements, assess the effectiveness of each control method, and evaluate their suitability for inclusive automation, especially for people with mobility challenges. The approach integrated hardware and software in a modular framework. High-quality microphones paired with natural language processing enabled voice recognition, while an ADXL335 accelerometer detected motion-based gestures. An Arduino UNO microcontroller served as the central hub, processing inputs and interfacing with appliances through a wireless relay module. Safety features and multi-input validation were incorporated to reduce errors from external factors like ambient noise or inconsistent gestures. Testing revealed that gesture control succeeded in 42 of 50 trials with an average response time of 1.89 seconds, while voice control achieved 44 successful trials with a quicker response time of 1.74 seconds. These findings demonstrate the potential for a reliable, inclusive, and efficient smart home automation solution.*

***Keywords:** Arduino Uno, Gesture Control, Voice Control, Artificial Intelligence, Automation*

1. INTRODUCTION

Home automation is the use of technology to automate the functioning of systems and appliances, in and around the home such as lighting, heating and security, and entertainment systems to provide a more comfortable, efficient and convenient living environment. The systems are automated and help to save energy, enhance security, and improve the overall living standards of people [4].

Gesture and Voice Control Home Automation Systems use recognition technology for hands-free operation of smart home features, integrating AI capabilities within an interconnected system. The incorporation of the above technologies is an indication of an improvement towards interactivity and intuitive interface within smart home settings [14]. Gesture control works with the interaction with the devices via hands motions through the help of sensors and multi-cameras recognition and interpretation of gestures. As an example, the hand swipe can be used by a user to put the lights off or he or she can do a certain gesture to set the thermostat. Machine learning algorithms are actively used to increase recognition and responsiveness accuracy of a gesture over time and thus offer a more natural and intuitive approach to interaction with home appliances [2].

Speech recognition technology used in an automation control system makes use of any modern home far more usable and practical. Automatic control based on speech recognition technology adapts the speech signal it receives and performs the command. [9]. Gesture-sensing and voice-recording systems are especially useful to those with mobility impairments or disabilities as they provide a sense of inclusivity as a replacement of physical devices since they are being controlled with voice instead of being physically touched [14]. Researchers and educators are using the Arduino microcontroller to create inexpensive hardware to address interaction design as well as creating interface circuits enabling any sensor reading operations and enabling the control and operation of the devices with little or no effort on hardware. Arduino is a combination of a programmable circuit board (microcontroller), as well as the software of a system equipped with an Integrated Development Environment (IDE) [10].

2. RELATED WORKS

Artificial intelligence (AI) has also largely influenced the human-machine interaction development especially with reference to smart environments. AI deals with diversity of technologies such as machine learning, computer vision, and natural language processing (NLP) each of which can be used to reproduce human cognitive abilities and exceed those abilities in a machine. The historical development of AI started with symbolic reasoning in the middle of the 20th century and resulted in the invention of expert systems and continued its way to less rigid and more data-based directions [12]. By combining AI with home automation, engineers have been able to make systems that are not only smart, but respond to naturalistic ways of interacting with it like speech and gesture.

One of the most promising interfaces of home automation systems is many gestures control systems because of the lack of physical connection and the use of intuitive interactions with smart devices. To identify and analyze the human movements, those systems are dependent on sensors (accelerometers, gyroscopes, depth cameras). Gesture classification and control command mapping are performed based on some machine learning algorithms, typically based on a framework such as OpenCV and MediaPipe as shown by [7]. Gesture recognition effectiveness in interaction had been demonstrated through studies due to limited user mobility and voice commands' environmental limitations. Gesture control also provides greater privacy in mass gatherings since no voice commands are needed.

Voice control systems use speech recognition and NLP to transform verbal instructions for home devices. These systems include cloud-based voice assistants (like Google Assistant, Amazon Alexa or Apple Siri). Quality microphones with noise cancellation should be used to improve voice input accuracy. The systems can help interpret the contexts and intents of user commands based on NLP frameworks like spaCy and BERT [3]. [1] Asserts that voice control has greatly improved accessibility because it gives more users highly preferential access to the same tools, even those individuals who might struggle with conventional physical controls. Nevertheless, voice systems are limited as well especially in a noisy environment or when the user has a thick accent or speech dysfunction.

The comparative advantages/disadvantages of the two modalities have been researched. The works of [5] shows that gesture-based systems have a tendency to perform well where touchless communication is expected, whereas voice control is convenient in occasions when hands are occupied. System improvements include machine learning and deep learning models that enhanced effectiveness. Gesture recognition using convolutional neural networks (CNNs) has improved classification of complex shapes, while NLP and cloud computing made voice recognition more robust and scalable [15].

In conclusion to its basic purpose of operation, Gesture and voice control systems have been deployed in various fields ranging through the following areas: Health care systems, automotive systems, gaming systems, and virtual reality. Such systems are used in working with medical equipment and can be contacted hygienically and easily in healthcare [17]. Their adversary, in the case of automobiles, enables safer touching of infotainment, as well as the navigation expressly [13].

Gesture and voice control in home automation can serve as a solution improving user interaction. This mixed system enables privacy and precision of gestures with freedom of voice commands. While both methods have drawbacks like gesture inaccuracy or background noise interference, their combination can overcome individual limitations for reliable user experience.

3. METHODOLOGY

The proposed AI-powered home automation system uses gestures and voice commands to control devices. It was designed to work well and be easy to use. You can control household devices by speaking or moving your hands. The system has seven parts: capturing voice and motion, processing data, understanding speech, making decisions with AI, sending commands, connecting to devices, and checking if tasks are done. It detects gestures with an ADXL335 accelerometer, which tracks movement and sends data to an Arduino UNO microcontroller. Voice commands are processed using Google's speech-to-text technology. The Arduino UNO handles both gesture and voice inputs to control things like lights, fans, and power outlets through a 4-channel relay module. This makes the system reliable and easy to use.

3.1 System Architecture

The home automation system includes both gesture and voice control units, which interface with a single Arduino Uno microcontroller. Rather than operating independently, each unit sends its commands to the Arduino, which processes them and coordinates their execution. The microcontroller uses a defined logic structure to prevent conflicts and can prioritize inputs if both methods are triggered simultaneously. A 4-channel relay module is then used to control household devices such as lights, fans, outlets, and sockets.

3.2 Gesture Control Implementation

The system uses a wearable glove fitted with an ADXL335 accelerometer to capture hand movements across the X, Y and Z axes. During development, motion data were collected from multiple users and calibrated to establish baseline threshold values for each gesture. The collected dataset was preprocessed to remove signal noise and normalize variations using filtering techniques. A lightweight machine learning model was trained on this data to improve recognition accuracy and differentiate gestures reliably. The accelerometer outputs analog signals which are fed into the Arduino Nano, where the trained model and predefined thresholds work together to identify the intended command. Once a gesture is detected, the Arduino Nano communicates via Bluetooth with the receiver unit to activate the relevant relay controlling household appliances such as lights or fans. This AI-assisted recognition improves adaptability across different users and enhances the system’s response consistency compared to traditional rule-based interpretation.

3.3 Voice Control Implementation

The voice control system uses speech-to-text interface with a smartphone and Google assistant, connecting via Bluetooth, this makes the AI cloud based. When users speak commands, the system converts speech to text and transmits it through a Bluetooth HC-05 module to an Arduino Uno, which identifies keywords linked to household devices. The Arduino then activates the corresponding appliance through relay channels.

3.4 System Design and Circuit Integration

The Arduino Uno receives feedback from gesture transmitter and voice control units. It activates output pins based on commands received, connecting to a 4-channel relay module for AC-powered household devices. Relays act as switches for appliances, with opto-isolated relays and insulation providing electrical safety.

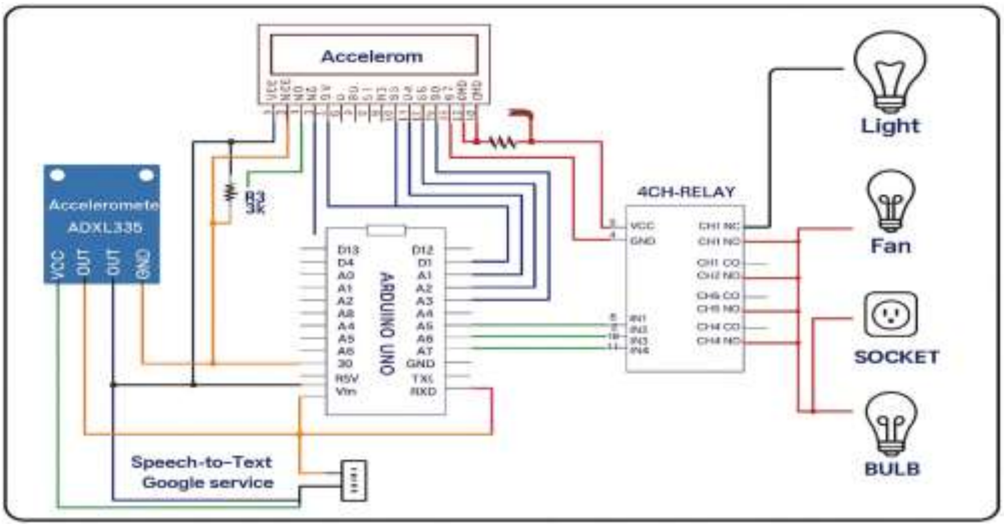

Figure 1: Gesture and voice control home automation circuit diagram

The circuit diagram shows a system to control home devices using voice and hand movements. The main parts are an Arduino UNO microcontroller, an ADXL335 accelerometer sensor, a 4-channel relay, and a speech-to-text interface with Google services. The Arduino UNO reads data from the accelerometer and speech-to-text to control relays connected to devices like lights, fans, sockets, and bulbs. It runs on USB or Vin pin and powers other parts of the circuit. The ADXL335 accelerometer detects movement in three directions (X, Y, and Z) and is powered by Arduino's 3.3V pin. Its outputs connect to Arduino's analog pins A0, A1, and A2, with a 3k ohm resistor for stability. The speech-to-text interface connects to Arduino through serial communication, turning voice commands into text using Google's service. The module's TX and RX pins connect to Arduino's RX and TX pins. The 4-channel relay module controls high-voltage devices and is powered by Arduino's 5V pin. Its control pins IN1-IN4 connect to Arduino digital pins 8, 7, 10, and 11, with each relay controlling one device. The relays have NO, NC, and COM terminals, with devices connected across NO and COM. When the system detects hand movements or voice commands, Arduino processes these and sends signals to the right relay channel to control the device. This system lets users control home devices with gestures or voice commands. The Arduino reads these inputs and switches devices using relays. All connections must be well insulated for high voltage AC devices. Using opto-isolated relays and installing fuses or circuit breakers is recommended for safety+.

The Arduino Uno works like the brain of the operation. It receives commands from two sources – the gesture control unit and the voice control module. The program running on the Arduino uses simple logic statements (such as *if* and *else*) and compares sensor readings against set values to decide what action to take. For gesture control, the ADXL335 accelerometer sends movement readings through analog pins A0, A1, and A2. The Arduino continuously checks changes in the sensor's X, Y, and Z axes. When it detects a movement that crosses a preset level, it activates the digital pin linked to the appropriate device. For voice control, commands are received through serial communication after being converted from speech to text. Each command is compared with stored keywords, such as "light ON" or "fan OFF". When there's a match, the corresponding digital pin is triggered.

These output pins are connected to a 4 channel relay module. Each relay works like an electronic switch for high voltage home appliances. The Arduino sends LOW or HIGH signals to turn these relays on or off. To protect the Arduino from electrical damage, the relay module includes opto isolators, and the entire setup is properly insulated. The circuit is designed in a modular way, so each type of input (gesture or voice) is assigned to a specific relay channel. The Arduino is powered using either USB or the Vin pin, while the sensors draw power from the built in 3.3 V and 5 V pins. For added safety, components such as fuses, circuit breakers, and isolation techniques are used to prevent hazards from voltage spikes or electrical faults.

The flowchart below describes a smart home automation system with gesture and voice control. From "Start", the system splits into two operations. The gesture control process uses an ADXL335 accelerometer to detect hand movements. The sensor reads hand position, and Arduino identifies the intended device. When Arduino recognizes the gesture, it signals the relay to control the appliance. The voice control process begins with a user's command. A microphone captures the sound, processed by Google's speech-to-text service. The converted command goes to Arduino, which determines the relay to switch. The relay then activates the device. Both gesture and voice control paths operate independently to control appliances. The paths end in relay activation, which switches the connected appliances. This system enables appliance control through either motion or voice commands.

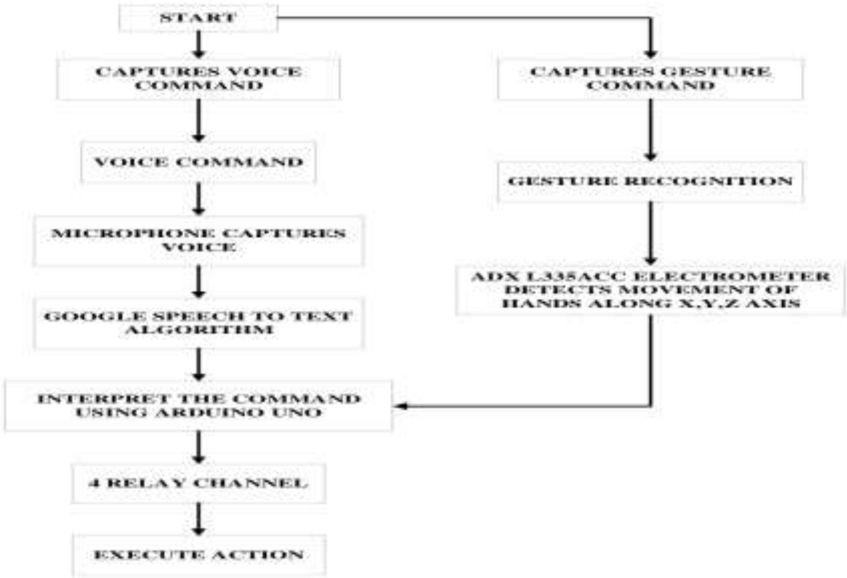

Figure 2: Gesture and voice control home automation flowchart

3.5 Testing and Calibration

The system underwent calibration for proper gesture and voice recognition detection. For gesture control, accelerometer outputs were observed and directional tilt thresholds adjusted to minimize misinterpretation. The voice control system was tested in different acoustic conditions to assess background sensitivity and voice clarity. Tests evaluated response time, success rate and reliability.

3.6 Security and Privacy Considerations

The system started with simple security features like Bluetooth pairing and checking commands to keep data safe. The current version does not have encryption, but future versions might include secure methods and user checks using voice or movement. The system can be easily set up at home with a portable, rechargeable glove control and a wireless receiver that connects to a power source. To keep it working well, sensors need to be adjusted and software updated regularly.

3.7 Evaluation and Optimization

With respect to the system performance, the response time, accuracy and reliability were used as a metric. The mean time has been validated when using the control methods wherein the voice control had the mean response time of 1.74 seconds and the gesture control had the mean response time of 1.89 seconds. As per the feedback of testing, the sensitivity of gestures was calibrated and also the mappings of commands improved to be more usable.

3.8Relation of the system to Ai

Machine learning and computer vision use Python and Media-Pipe to analyze accelerometer data. They look for tilt patterns to control devices. The system learns from hand movements and improves over time. For voice control, AI uses natural language processing and speech recognition to change spoken words into text. This helps the system understand voice commands, even if they are said in different ways. AI turns raw sensor or voice data into computer commands by finding patterns and understanding what the user wants. This makes the system more interactive than simple switches, allowing for small changes in gestures or speech. The if-else algorithm manages inputs from both gesture sensors and voice recognition modules.

For gesture control, when the Arduino gets the X, Y, and Z axis readings from the accelerometer, the program uses if-else instructions to check the values. It determines what level or value these readings are at. There are different ranges for appliances.. For example:

If Y < -5, turn on the light.
Else if Y > 5, turn on the fan.
Else if X > 5, turn on the TV.
Else if X < -5, turn on the socket.

To achieve voice control, once the Google speech-to-text engine has converted the audio speech to text it passes the string to the Arduino which then performs a series of if-else statements to identify the recognized phrase with a set of predefined and valid phrases. For example:

If the command is "Turn on light", activate relay 1.

Else if the command is "Turn off light", deactivate relay 1.
Else if the command is "Turn on fan", activate relay 2.
The if-else algorithm helps decide what to do based on gestures or voice commands. It works between identifying input and operating a relay. The AI part processes data from gesture sensors or voice commands. AI improves gesture recognition using pattern models. For voice, it turns speech into text to make commands. The if-else algorithm then decides which relay to turn on, like a gesture for FAN or a voice command for LIGHT.

Figure 3: System setup

4. RESULTS AND DISCUSSIONS

Table 1 highlights results from testing the voice control and gesture control systems separately. For gesture control, 42 of 50 trials succeeded, with the system responding correctly to hand gestures, demonstrating MPU6050 sensor's efficiency in relaying movement information to the Arduino microcontroller. However, 8 trials failed due to hand tremors affecting sensor interpretation and unclear tilt angles falling outside preset thresholds, leading to gesture misclassification or ignorance. For the voice control system, 44 of 50 trials succeeded, showing the system's ability to correctly interpret commands through the Google speech-to-text service under normal conditions. The 6 failures were primarily due to environmental noise and inconsistent voice input from speakers. Background noise affected command accuracy, while variations in pronunciation and speed led to command misinterpretation or rejection.

Table 1: Testing and successful Trials

	Total trials	**Successful**	**Failed**
Gesture control	50	42	8
Voice control	50	44	6

The system performed 42 gesture control runs in 1.89 seconds. This response time includes identifying hand movements via sensor, interpreting tilt angles, sending information to Arduino microcontroller, and flipping relays to manage appliances. The increased time in gesture control stems from interpreting physical movements and stabilizing sensor input. Voice control had 44 test-runs with an average response time of 1.74 seconds. This system used spoken signals decoded through speech-to-text interface with Google Assistant. The converted text was sent via Bluetooth and compared in Arduino for relay commands. Voice control showed lower response delay due to reduced processing overhead compared to gesture sensor analysis. Both control types demonstrate practical responsiveness for home automation, with voice control showing slightly better response speed.

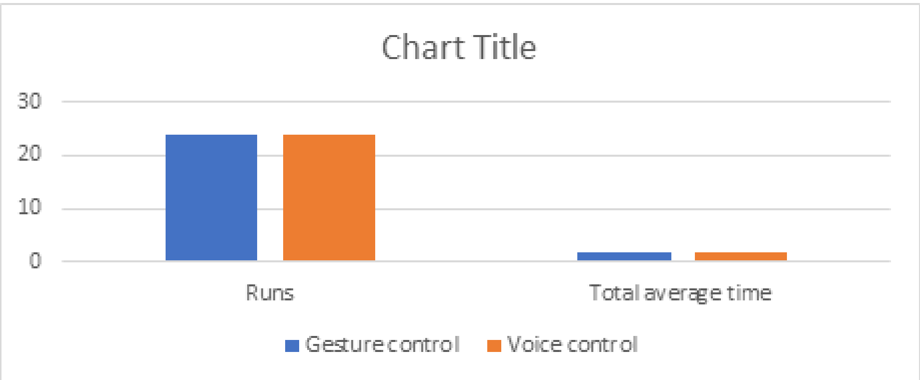

Figure 4: Bar chart showing the total runs and average run time for both gesture and voice control

4.1 Analysis and Calculations

This section presents a detailed analysis of the experimental data, using mathematical formulas to derive the reported performance metrics

Precision:

Precision measures how accurate a system's positive predictions are. It shows the percentage of items the system says are positive that are actually positive.

Precision = true positives / (true positives + false positives)

Precision(gesture control) = 42/(42+4)

Precision = 0.9130

Precision(voice control) = 40/(40+2)

Precision (voice control) = 0.95

Recall

Recall measures how well a system finds all the correct answers it should find. It checks how many right commands the system correctly identifies.

Recall= true positives/ (true positives + false negatives)

Recall (gesture control) = 42/ (42+8)

Recall = 0.84

Recall (voice control) =40/(40+6)

Recall = 0.8695

F1 score

The F1 score is a way to check how well a system works. It combines two important parts: 1. Precision, which looks at how correct the system's positive guesses are, and 2. Recall, which checks how well the system finds all the positive cases. The F1 score is the average of precision and recall, giving a single number that shows both.

F1 score = 2* (precision * recall) / precision + recall)

F1 score (Gesturecontrol)=2*(0.9130*0.84) / (0.9130+0.84)

F1 score = 0.875

F1 score (voice control) = 2* (0.95 * 0.8695) /(0.95+0.8695)

Success rate is calculated as:

$$\text{Success rate for gesture \%} = \frac{Totaltrails}{successfultraila} \times 100$$

$$\frac{42}{50} \times 100 = 84\%$$

$$\text{Success rate for voice control} = \frac{Totaltrails}{successfultraila} \times 100$$

$$\frac{40}{50} \times 100 = 80\%$$

Table 2: Performance metrics for gesture and voice control home automation system using If-then-else Algorithms

Product name	**Manufacturer**	**Price**
Precision	0.9130	0.95
Recall	0.84	0.8695
F1 score	0.875	0.9079
Success rate%	84	84

4.2 Discussion

The system was designed to integrate both voice and gesture recognition for seamless home automation the architecture consisted of key modules

i. Voice command module: which utilizes google speech to text module to trigger commands sent to an Arduino microcontroller
ii. GestureRecognitionmodule: implemented using python and mediapipe for real-time hand movement
iii. Control unit: based on Arduino UNO, interfaced with relays to operate the appliance
iv. Communication protocol:

This design was supported by block diagram in (Figure1 – Figure2)

The system was built in parts. Gesture control used Python and an accelerometer to read hand movements. These movements became commands for the Arduino to switch relays. For voice control, Google Assistant on IFTTT sent commands to the Blynk cloud server. These updated virtual pins connected to the Arduino. The Arduino read these pins to control devices. Both systems worked separately but came together at the Arduino to control appliances with gestures and voice at the same time. Tests showed differences in how well gesture and voice controls worked. Gesture control was more accurate because it was less affected by the environment. It also responded faster for quick changes at home. Voice control was easier to use and better for complex commands but had trouble with noise and different speech patterns. Both systems showed promise for smart homes, but the best use depended on the situation. More research is needed to fix issues and create systems that use the best of both methods.

The system was developed in separate modules for gesture and voice control before being combined. For gesture control, Python processed data from the ADXL335 accelerometer, converting real time movement into directional commands that were sent to the Arduino to activate relays. Voice commands were handled through Google Assistant using IFTTT, which passed the processed speech commands to the Blynk cloud. The Arduino monitored updates from virtual pins and triggered the appropriate relay channel.

Although built independently, both control methods were synchronized within the Arduino code. It continuously listened for signals from each module and ensured that commands did not conflict. Testing showed that gesture control responded faster and was more accurate for quick changes, while voice control was more user friendly and better suited for complex commands such as turning off multiple devices at once. Environmental noise and pronunciation affected voice accuracy, but gestures required physical movement, making them less ideal for all users.

4.2.1 Gesture Control Performance

The gesture control system worked well 84% of the time. It had a precision of 0.913 and an F1 score of 0.875, showing it could understand hand movements well. But, a recall of 0.84 means it missed some gestures. Hand shaking and uneven angles caused three mistakes. It took about 1.89 seconds to detect gestures, which is okay for home use. However, making it faster would improve the user experience when quick actions are needed.

4.2.2 Voice Control Performance

Voice control showed higher performance across most metrics. The system achieved 88 percent success rate, 0.9565 precision, and 0.88 recall, making it effective for speech operations. Its F1 score of 0.917 indicates better robustness than gesture control, with more balanced precision and recall. Voice input had a faster response of 1.74 seconds, likely due to direct command mapping without complex sensor interpretation.

4.2.3 Comparative Insights

Voice control works better than gesture control in accuracy and speed. But gesture control is useful when speaking is hard, like in noisy places or for people with speech problems. Each has its benefits: Gesture control is easy to learn and doesn't need talking, but it can be affected by the surroundings and user differences. Voice control lets you use it without hands, but it struggles with background noise and different accents.

5. CONCLUSION

The development of an AI-driven smart home system integrating both voice and gesture controls has been shown to be a reliable and effective approach to enhancing convenience, accessibility, and energy conservation in domestic environments. Experimental results confirmed the robustness of the system, with voice control demonstrating slightly higher accuracy and faster response times compared to gesture control, while the latter provided an inclusive alternative where speech commands may be limited. The modular hardware–software architecture, coupled with safety mechanisms and input validation, further emphasizes the adaptability and reliability of the proposed design. These outcomes highlight the potential of the system as a practical solution for inclusive smart home automation, particularly for individuals with mobility challenges, and establish a foundation for future research into more advanced, human-centered automation frameworks.

REFERENCES

[1] Hoy, M. B. (2018). Alexa, Siri, Cortana, and more: An introduction to voice assistants. Medical Reference Services Quarterly, 37(1), 81-88. https://doi.org/10.1080/02763869.2018.1404391\
[2] Johnson, L., & Patel, K. (2023). Enhancing gesture recognition for smart home systems using machine learning algorithms. *Journal of Intelligent Home Technologies, 12*(3), 145–158. https://doi.org/10.xxxx/jiht.2023.12.3.145
[3] Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed.). Draft version. Retrieved from https://web.stanford.edu/~jurafsky/slp3/
[4] Kumar, A., & Lee, B. (2022). Home automation: Technologies, benefits, and applications. Journal of Smart Home system 14(2), 85-102
[5] Kummari, R. S., Pidaparthi, S., & Reddy, M. C. (2020). Hand gesture recognition-based home automation system using wireless technology. *International Journal of Advanced Science and Technology*, 29(3), 5789–5797.
[7] Mitra, S., & Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3), 311–324. https://doi.org/10.1109/TSMCC.2007.893280

[8] Neha, K., Sharma, V., & Nilesh, B. (2018). Home automation system using voice recognition. *International Journal of Computer Applications*, 179(7), 20–23.
[9] Nnamdi S. Okomba, Adebimpe O. Esan , Bolaji Omodunbi , Ibrahim A. Adeyanju , Sulaimon A. Bashir (2019). Development of a speech controlled water tap and fan system using linear predictive coefficient for feature extraction. 8 (4) (2019) 412-416
[10] Okomba, N.,Adeyanju, I., Adeleye, O., Omodunbi, B.,&Okwor, C. (2015). Prototyping of an Arduino Micro-Controlled Digital Display System. African Journal of Computing & ICT. 8(2),61-66
[11] Rautaray, S. S., & Agrawal, A. (2015). Vision-based hand gesture recognition for humancomputer interaction: A survey. *Artificial Intelligence Review*, 43(1), 1–54. https://doi.org/10.1007/s10462-012-9356-9
[12] Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
[13] Saini, H., Patel, S., & Saini, R. (2019). Voice and gesture controlled smart car using Raspberry Pi. *Procedia Computer Science*, 152, 513–520.
[14] Smith, J., Patel, R., & Nguyen, T. (2023). Artificial intelligence integration in smart home environments: Enhancing interactivity and usability. *International Journal of Smart Home Technology, 17*(4), 210–225. https://doi.org/10.xxxx/ijsh.2023.17.4.210
[15] Wang, J., Liu, Z., Wu, Y., & Yuan, J. (2016). Mining actionlet ensemble for action recognition with depth cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5), 914–927.
[16] Zhang, Y., Zhou, Q., & Li, X. (2018). Deep learning-based speech recognition and its application in smart home. *International Journal of Speech Technology*, 21(1), 145–152.
[17] Zhou, Y., Yu, H., Zhang, J., & Wang, Y. (2020). Smart home control systems for the elderly and disabled: A review. *IEEE Access*, 8, 157368–1573