SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I4P107

Volume 12 Issue 4, 75-83, April 2025

Original Article

Innovative Approach to Smart Home Automation: Leveraging Hand Gesture Recognition for Enhanced User Interface and Control

Abhishek Madankar[1], Minal Patil[2], Anshul Pardhi[3], Manoj Lahudkar[4], Shravani Duragkar[5], Shital Telrande[6]

[1,2,3,4,5]*Electronics & Telecommunication Engineering, Yeshwantrao Chavan College of Engineering, Maharashtra, India.*
[6]*DMIHER, Wardha, Maharashtra, India.*

[1]*Corresponding Author : vicky.madankar123@gmail.com*

Received: 04 February 2025 Revised: 06 March 2025 Accepted: 08 April 2025 Published: 29 April 2025

Abstract *- Gestures are a revolutionary assignment geared toward revolutionizing the use of electronic gadgets in human-machine communication. Harnessing the strength of gestures, this assignment seeks to create simple and bendy interactions that put off the need for conventional body manipulation. With computer ideas and superior predictive and machine-gaining knowledge of techniques, the gadget acknowledges and accurately translates a wide variety of gestures. This gesture can then be mapped to important commands, permitting clients to control electronic devices in a fantastically simple way. This technology of power programs is available in all sizes, from clever homes to industrial structures. Providing a fingerless and green manner to interact with devices, this application has the ability to enhance existence and productivity in a fulfilling and obvious manner. Future mastering guidelines encompass exploring greater complex famous gestures, integrating voice commands, and offering greater complex real-time structures for customers to study quite simply.*

Keywords *- Pi Camera, Raspberry Pi 4, Relay, VNC.*

1. Introduction

The system is novel in controlling electrical appliances through hand gestures. It provided a hands-free and simple way to interact with technology. The main system contains a Raspberry Pi 4, a small but powerful single-board computer that handles the data processing and command execution from multiple sensors. The main input device in this system is PiCamera, which takes real-time snapshots of user hand gestures. The Raspberry Pi then processes these gestures using computer vision and performs the opposite action of that gesture, e.g., switching on or off appliances. It connects the Raspberry Pi with external components such as the camera, relay module and power supply, known as the I/O Interface.

The article explores using vision-based approaches integrated with machine learning algorithms for real-time hand gesture recognition. It consists of a modular architecture that includes data acquisition, preprocessing, feature extraction, and classification. By utilizing centroid distance values and Support Vector Machines (SVMs), the system achieves robust posture recognition with a reported accuracy of 99.4% for static hand gestures and 93.72% for dynamic gestures. The focus is designing user-independent systems that address challenges like visual noise, incomplete information, and scalability. A notable feature of this research is its emphasis on the flexibility and adaptability of the system for various applications, such as robotic control and sign language recognition. The study also introduces the Referee Command Language Interface System (ReCLIS) for robotic soccer refereeing and a Portuguese Sign Language recognition prototype. Both implementations highlight the system's capacity to operate under controlled environments, emphasizing computational efficiency and error tolerance.

Despite these achievements, the authors acknowledge the limitations in handling dynamic environments and continuous gestures. They suggest future improvements in gesture modeling and machine learning techniques to enhance system robustness and scalability, paving the way for broader real-world applications [1]. The proposed article provides a detailed review of vision-based hand gesture recognition systems from 2014 to 2020, focusing on their methodologies, challenges, and future directions. It categorizes gesture recognition systems into static and dynamic gesture recognition, emphasizing their applicability in real-world scenarios such as sign language translation and Human-Computer Interaction (HCI). It highlights critical components of gesture recognition, including data acquisition, feature extraction, and classification. One of the key insights is the disparity in recognition accuracy between signer-dependent and signer-independent systems. While signer-dependent systems achieve accuracy levels of up to 98%, signer-independent systems struggle with variances, achieving an average accuracy of 78.2%. This discrepancy underscores the importance of creating robust systems capable of handling diverse user input and environmental conditions.

The review identifies significant challenges, including handling uncontrolled environments, segmenting hand gestures accurately, and addressing non-gesture movements such as coarticulation and movement epenthesis. These challenges affect the system's ability to detect continuous gestures accurately. Most reviewed systems rely on controlled laboratory conditions, which limits their practical applicability. Moreover, the research highlights the need for better datasets that capture dynamic gestures in naturalistic settings. To address these limitations, the paper advocates for advances in feature extraction methods and adaptive algorithms, emphasizing the potential of machine learning and deep learning techniques to improve scalability and robustness.

Future research is encouraged to focus on creating signer-independent models and systems that work seamlessly across varying environments [2]. The application of transfer learning methods and convolutional neural networks for recognizing American Sign Language gestures and the general implications of sign language as a tool for non-verbal communication, particularly with a perspective towards deaf and hard-of-hearing individuals and persons with other forms of disability, is proposed. The author introduces a vision-based hand gesture recognition wherein a thresholding technique is employed as a preprocessing to make the recognition invariant to skin, lighting, and other differentiating factors. The authors make a panel to compare their methods with past studies that have ever been conducted explaining the shortfalls of earlier systems such as HMM and Naïve Bayes Classifiers, which are monoperformatic since they define certain gesture performance parameters in an overlapping manner.

In the case of static images, where hand silhouette mouth shape solidly mapped to one out of 26 gestures representing letters of the English alphabet, the proposed CNN model managed to achieve 99.96% recognition accuracy. It addresses how the variations in user-defined gestures, the zone in which they are performed, and noise can affect the response of existing gesture recognition systems and explains how the future of gesture-based communication systems using machine learning can be made more user-friendly and adaptive [3]. The development of HGR systems, specifically targeting user-efficient personalized gestures and the deficiencies related to inter-user differences, data generation and recognition in real life, is given in the proposed article. The role of HGR is crucial for HCI in the fields of virtual games, robot manipulation, and sign language conversion. The so-called prior works in the field can be divided into two major groups: static and dynamic gesture recognition, which places emphasis on problems such as gesture boundary detection, varying lighting conditions, and occlusions. Deep convolutional networks, as well as multimodal approaches, have been promising measures in the field of enhancing the accuracy levels achieved so far. Also mentioned are the methodologies of gesture recognition themselves, including contrastive learning and lightweight Multilayer Perceptrons (MLPs). Earlier studies, in most cases, focused on large databases or multi-camera solutions; this paper offers an interface allowing users to create behavioral gestures with low data levels using Mediapipe for real-time tracking.

This innovation should improve the system's usability as it focuses on individual styles of gestures, leads to lesser training and enhances the practicality of the system [4]. Massive advancements have been made in the literature on Hand gesture recognition, mainly aiming at improving Human-Computer Interaction (HCI) through effective and natural interfaces. HGRs are of utmost importance for static gestures, especially for sign language interpretation, remote control, and gaming interfaces. Traditional HGR approaches, such as skin detection systems and multiscale color features, suffer from accuracy and robustness, especially in more complex backgrounds and changing illumination. Advanced recent developments are posed on estimation techniques such as High-Resolution Networks, which did well in detecting human key points for critical information regarding gestures without requiring wearable sensors. Its multi-resolution architecture of processing has been proven for high spatial accuracy applications.

Another use is MobileNetV2, which is aimed at applications on resource-constrained devices and is widely used in HGR tasks to extract spatial features with minimal computation. Current research focuses on efficiently and adaptively combining multiple data streams, such as key points and hand-region bounding boxes, to enhance recognition and robustness across environments and datasets.

Additionally, by applying two-pipeline architectures, recent HGR systems can process both image and keypoint data to make better recognition under various light conditions and complex backgrounds with considerable potential for real-world application and edge deployment [5]. To achieve a better hand silhouette, the study integrates depth maps while the gesture is represented using a multimodal hybrid descriptor containing both boundary- (Fourier descriptors and curvature features) and region-based (moments invariants) features. Gesture recognition is based on an ensemble of one-vs-all SVMs and demonstrates results that are quite satisfactory considering publicly available datasets. The findings of the research indicate a considerable promise of vision-based hand gesture recognition for applications like HCI, sign language interpretation and Virtual Reality environments. As there have been some advancements, the authors highlight the existing problems that persist with regard to real-world applications and, more specifically, to uncontrolled environments with rich backgrounds, thus ensuring that there is plenty of room for future developments in this area [6]. The improvement of hand gesture recognition systems, with emphasis on the benefits they offer in Hand-Computer Interaction (HCI) as these allow the users to control the machine with hand movements instead of using any input devices, is described in the current article. The article focuses on the steadily increasing role of hand gesture recognition in such fields as robotics, virtual gaming, sign language interpretation and HCI. There have been a number of such investigations, including the recognition of static and dynamic gestures. Static gestures are less complex and process less computational resources, so they are used in restricted settings, whereas dynamic gestures, which are more involved, perform the sequence of movements and are more suitable for real-time settings.

The research work provides a flow chart highlighting three processes in gesture recognition, which include: image segmentation, feature extraction and classification. Putting all together, image segmentation constitutes isolating the area of the hand in the input image from the background image. This is done by applying skin color bounding boxes or dynamic tracking. The shape of the hand, its palm's center and fingers' tips are parameterized on their positions to aid in feature extraction, which is very important in recognition accuracy. Finally, advanced machine learning approaches like CNNs, which focus on specific classification features, are utilized to achieve the classification objectives [7]. The literature defines textual inputs and gestures as having developed much, focusing on improving human-computer interaction through various recognition methodologies.

Vision-based techniques are mainstream, allowing gesture recognition using cameras and image processing algorithms. Relevant methods in this area include segmentation techniques, using RGB color spaces, depth sensing, and real-time tracking. Other approaches entail Convolutional Neural Networks (CNNs) for deep learning, which have been found to work well with gesture classification because they can handle complex patterns and obtain a highly accurate outcome. Some commonly used depth-sensing cameras use models such as Kinect and Hidden Markov Models (HMM) that are widely applicable in dynamic gesture recognition.

For example, webcam-based recognition permits intuitive interaction by carrying out hand gesture recognition without physical contact with the device, thereby improving user experience. Other approaches have used accelerometer and gyroscope sensors in gloves, which are more accurate in recording motions but less convenient. Some researchers have further explored real-time techniques and even systems using low-cost cameras to suit more accessibility.

The field continues to deal with challenges that include recognition accuracy in different lighting conditions, occlusion, and user independence, which can make gesture recognition systems more robust and scalable toward wider applications in daily interaction technology, robotics, and assistive devices [8]. A comprehensive overview of recent advances in hand-gesture recognition systems. Such systems, it argues, are gaining more importance in applications such as HCI, robotics, and sign language interpretation.

The paper categorizes the recognition process into three major stages: gesture segmentation, feature extraction, and classification. It discusses different methodologies such as skin color models, neural networks, and clustering algorithms with a focus on advantages as well as problems. Applications are discussed along with the analysis of techniques for the recognition, such as Hidden Markov Models (HMMs) and Neural Networks (NNs). The review addresses certain common problems, such as sensitivity to lighting and computational complexity, and summarizes the results of the key studies through comparative tables.

Conclusively, it evaluates the suitability of methods for specific applications and identifies areas for further research [9]. It primarily zeroes in on the recognition stages of image acquisition, segmentation, tracking, and classification, which together form the basis of vision-driven hand gesture recognition technology. Various recognition techniques are discussed in the paper: sensor-based, vision-based, and deep learning methods. Among the key challenges for an effective real-time hand gesture system is separating the hand from its background under varying lighting and occlusion conditions, which impacts accuracy and usability. These methods fall into a few categories.

Some are color and silhouette-based models; others are contour-based or mainly described through machine-learning approaches, such as CNNs, HMMs, and LSTM networks. Each of these has unique advantages: sensor-based ones tend to be more accurate but need wearable devices, while vision-based ones achieve user-friendly interaction in terms of use but suffer from configuration complexity.

Resilience, scalability, and user independence are the keys to enhancing a system's robustness in various environments and users. The review aggregates various studies to point out gaps and call for improved algorithms that address current system inadequacies, thereby suggesting the need for real-time adaptable and efficient gesture recognition solutions [10-13].

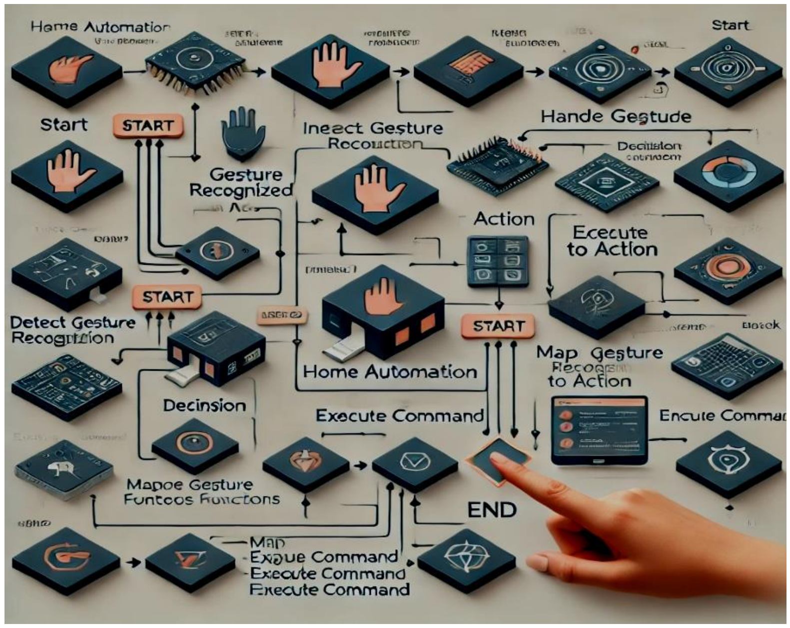

Fig. 1 Home automation system

2. Literature Review

The author uses the Raspberry Pi module 3B+ to implement a home automation system using voice and gesture control [14]. The main drawback of the Raspberry Pi module 3B+ is it is very slow. The time consumption for processing the image is very high. In this research, Raspberry Pi module 4 is used, which is very fast. It quickly senses the input from the image and operates the load accordingly. Raspberry Pi 4 is 340% faster than Raspberry Pi Module 3B+. Raspberry Pi 4 uses a 1.5 GHz processor, while Raspberry Pi Module 3B+ uses a 1.2 GHz processor. Raspberry Pi 4 is used for multitasking purposes. Hand gesture tracking and movement are proposed in this article. These features are not specified by the article [14]. The features used in the proposed work differ from those in the article [14]. The proposed system is cost-effective compared to the other models because very few components are used in this design. The system is compatible with the latest systems like Amazon Echo. The system additionally provides a hand gesture control facility for the people of all edge groups.

3. Methodology

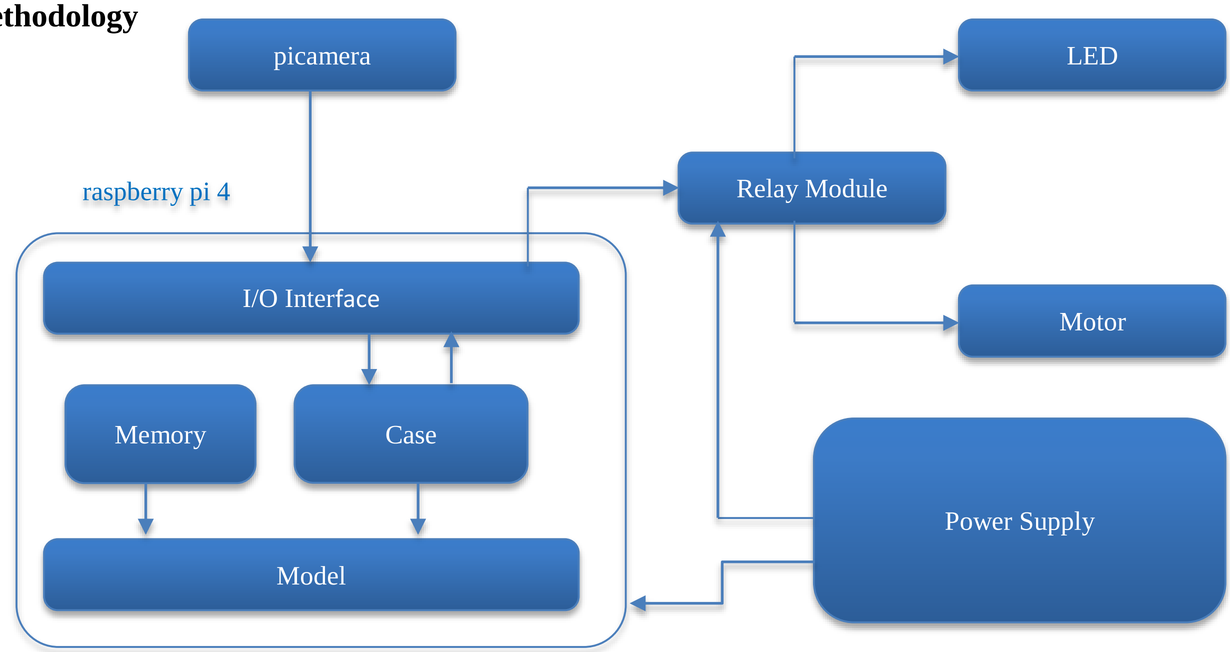

Fig. 2 Block diagram of the proposed system

The above block diagram depicts a more advanced system that employs hand gestures to control electrical appliances. This system works thanks to various devices that interpret hand gestures and use them as commands to control specific machines. Below, I will illustrate the several components that are fundamentally necessary for this system and show how they combine to give rise to a complete system.

3.1. The Raspberry Pi 4

The most central and most powerful single-board computer, Raspberry Pi 4, is placed at the core of the system. It acts as a Central Processing Unit (CPU) in the system that commands and manages communication among different portions. Next, let us decompose the important features of the Pi 4.

3.2. CPU (Central Processing Unit)

As for the Raspberry Pi 4, the device's CPU will refer to the system as the brain. Executing the commands that are in the system's memory and carrying out the operations required with the information received from the separate input and output devices connected to the Raspberry Pi is the role of the CPU as well. The CPU performs the calculations based on the predefined logic and interprets the hand movements the camera recognises to manipulate the devices. Such high capabilities of the Raspberry Pi 4 enable this processing to be accomplished within the appropriate time scale, which is important to providing real-time interaction with the system.

3.3. Memory

Memory is, in another way, the same as Sensory storage for the peripheral components or devices of the system, the Raspberry Pi 4, and interfaces. It temporarily holds the commands and context necessary for the system to work accurately. When a gesture is identified, the CPU processes information that is later written into the memory for fast access in the future. This is very useful as the system’s response time is reduced and allows the controlled devices to be commanded almost instantaneously.

3.4. The I/O Interface

Communication is the basic function of interaction with the external world, so it is essential that there exists a strong link between Raspberry Pi 4 and the external systems. The I/O interface enables the ports and connections required to interface components like cameras, relay modules and power supply units. The I/O interface also provides an input for signals from the peripheral devices to the Raspberry Pi for output signal processing.

3.5. Camera

Hand gestures are captured by a camera connected to the I/O interface, which is vital. It is possible to utilize the PiCamera, which is a dual camera that has been designed for Raspberry devices to output high-definition images or videos that will later be processed in the systems to identify hand movements.

3.6. Relay Module and Power Supply

These devices also interface with I/O. The active relay switch module, used by the Raspberry Pi, is used to control electric devices that are turned on or off. The power supply serves as a source for the modules that are connected to the Raspberry Pi and even the Raspberry devices themselves.

3.7. PiCamera

In this system, the PiCamera module is essential for capturing the user’s hand gestures. The high-definition camera is made specifically for Raspberry Pi and connects to the device’s GPIO pins or the camera interface. The camera takes several pictures or videos of the user’s hand gestures, which are further processed by the Raspberry Pi towards a specific motion or pose.

3.8. Hand Gesture Recognition

The user’s hand movements are tracked by the PiCamera, which captures the image and sends it to the Raspberry Pi for processing. The system does eye tracking by recognizing the target and computes these hand gestures using computer vision techniques once a target is tracked. The system once identifies a hand gesture that performs an action such as a light switch on or off or fan temperature depending on which command was programmed for which particular hand sign.

3.9. I/O Interface (Communication Hub)

The I/O Interface is a critical component, acting as a system's communication centre. It connects varied modules and provides a channel for data flow between them. This piece is of the essence since it stands between the Raspberry Pi and all external devices, ensuring their work gets coordinated as parts and pieces of a system.

3.10. GPIO Pins

The General Purpose Input/Output (GPIO) pins are typically how the Raspberry Pi interfaces with external hardware like sensors, cameras, or actuators. Essentially, these pins enable the system to send and receive signals to control attached components, like a relay module that switches appliances on or off. The mapping and transmission of these signals fall under the I/O Interface to guarantee that the appropriate actions are taken based on the user's hand gestures. Incorporating the important connections between components is achieved through the I/O Interface. Connecting the PiCamera to Raspberry Pi for image data processing and then linking the relay module to receive control signals for appliance management falls under the I/O Interface. Managing all these relationships is another responsibility of the interface to ensure proper working of the whole system. This I/O Interface also ensures that modules with equivalent GPIO connections are able to work together in harmony. For instance, if two or more sensors or actuators are to use the same set of GPIO pins, proper routing and data transmission handling done by the I/O Interface avoids any conflicts, assuring you that the system will work as required.

3.11. Relay Module and Electrical Appliances

The Relay Module is the most important part of that system because it is an intermediary switch that controls all electrical appliances. When a specific hand gesture is detected and interpreted by Raspberry Pi, a signal is sent to the relay module through the I/O Interface. Then the relay module will then activate or deactivate the connected appliance based on the recognised gesture. This will allow users to control appliances, for example, lights, fans, or air conditioning systems, through very simple hand gestures and thus make this system both practical and efficient.

4. Flow Chart

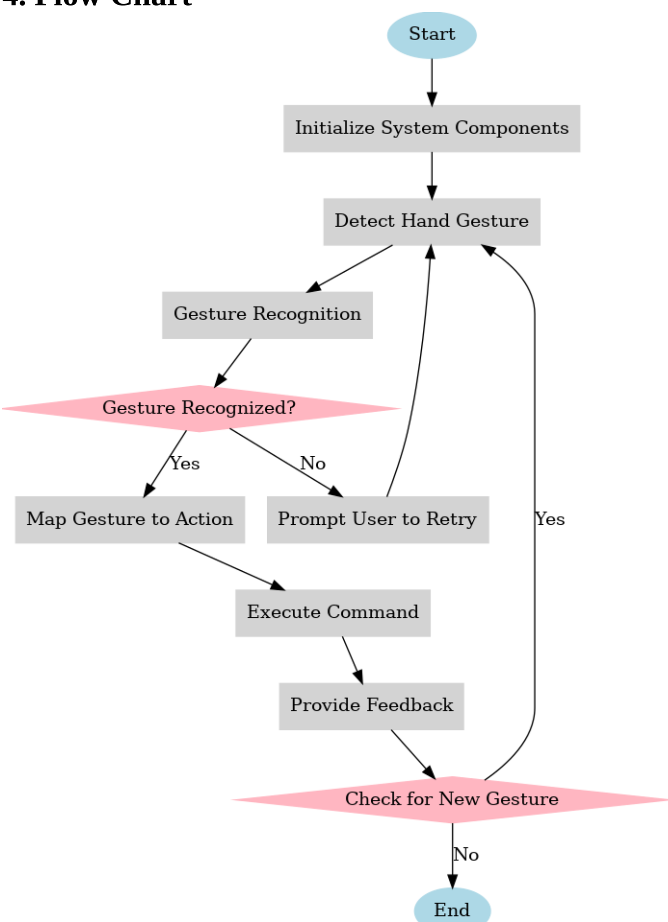

Fig. 3 Flowchart of system works

The system begins its operation. Initialize System Components Sets up all necessary components, including gesture sensors (e.g., cameras or Leap Motion controllers) and processors (e.g., microcontrollers, computers). Loads a pre-defined gesture library for recognizing specific commands. Detect Hand Gesture: The system uses sensors to capture hand movements or images of gestures. Prepares the input data for analysis. Gesture Recognition: Processes the captured gesture using algorithms (e.g., machine learning or heuristic-based).Matches the detected gesture with a library of known gestures. Gesture Recognized? (Decision Point) Yes: If the gesture is recognized, the system maps it to a corresponding action. No: If the gesture isn't recognized, the user is prompted to retry, and the process loops back to detecting gestures. Map Gesture to Action: Translates the recognized gesture into a specific command, such as turning on/off a light, adjusting a thermostat, etc. Execute Command Sends the command to the appropriate home automation device, which performs the desired action. Provide Feedback: Confirms the command execution through visual (e.g., LEDs), audio (e.g., beeps), or other feedback mechanisms, ensuring the user knows the action is completed. Check for New Gesture (Decision Point):Yes: If the system detects a new gesture, it loops back to detecting the gesture.No: If no further gestures are detected or a shutdown signal is given, the process ends. The system stops, concluding the home automation operation.

4.1. Key Features

4.1.1. Decision Points

The system checks if a gesture is valid or if further gestures are detected, making it interactive and dynamic.

4.1.2. Feedback Mechanism

Ensures the user is aware of successful command execution.

4.1.3. Looping Structure

Allows the system to operate continuously, detecting and responding to multiple gestures.

This flowchart encapsulates the main logic for a gesture-controlled home automation system, offering an intuitive and efficient way to manage smart home devices.

5. Working

A system for controlling electrical appliances through hand gestures would include some hardware and software parts. Hardware Typically includes the following:

5.1. Raspberry Pi

This is one single-board computer that is very versatile and acts as the brain of the system. It processes the images captured and controls the relay module.

5.2. Camera Module

This module will capture live feed video of the hand gestures made by the user.

5.3. Relay Module

This particular module will work as a switch. With the help of this module, the Raspberry Pi can turn on the power supply for electrical appliances.

5.4. Electrical Appliances

These are the devices to control. This may include lights, fans, and all other electronic devices. The software section is mostly Python code, using commonly used libraries like OpenCV and RPi.GPIO.

The camera module continuously captures video frames. These frames are processed using OpenCV, which includes techniques such as skin color detection and shape analysis, which enables the identification of the hand. Features like the positions of fingers and hand orientation are then extracted from the identified hand region. For gesture classification, the detected gesture may use pre-trained models. The algorithms scrutinize the extracted features to determine the intended gesture. Some typical gestures include waving an open hand, making a fist, or even specified finger positions. Raspberry Pi generates appropriate signals for the identified gesture. The signals are sent to the GPIO of the Raspberry Pi, which is connected to the relay module. The relay module receives the signal and switches on and off the power supply going through it to the electric appliance; that is a key challenge and consideration: It should be able to process images and recognize gestures in real time. Appropriate algorithms and optimization techniques must be applied to achieve this performance level. Takes into account the nature of lighting conditions because variations in lighting conditions are likely to propagate errors in hand detection and feature extraction. Consequently, adaptive thresholding and color normalization are employed to alleviate the effects of such variations. The camera shall be calibrated for perspective and gesture recognition. Noise and interference between the relay module and the Raspberry Pi may be a result of other external factors. Shielding, proper wiring, etc., are done to curb these problems. To enhance the user experience, a user-friendly interface can be employed, which may include a visual depiction of the recognized gestures or a system that provides audio feedback when gestures are made. Working as described above, these problems can be solved and further enhanced when developing this system, thus enabling the development of a consistent, reliable, and more intuitive hand interface-based home automation system. This technology allows us to change the ways we interact with our surroundings. Our lives become highly convenient as well as accessible.

6. Result

It includes a combination of hardware and software that work in conjunction with each other. Smart home control through selectional gesture recognition Hardware consists of a Raspberry Pi with a camera module that reads the hand gesture and a relay module that can switch appliances on and off. Initially, the relay switches on the electrical appliances such as fans, lights, etc. Software details: They used Python, OpenCV skin colour detection, shape analysis and RPI GPIO for switching relay. These techniques categorize hand gestures using previously learned models, with the finger positions as a feature obtained from images. Even simple techniques such as adaptive thresholding will be required in order to detect gestures accurately in real-time video, and noise reduction must also be applied through the code. That makes automating the home much easier and more flexible.

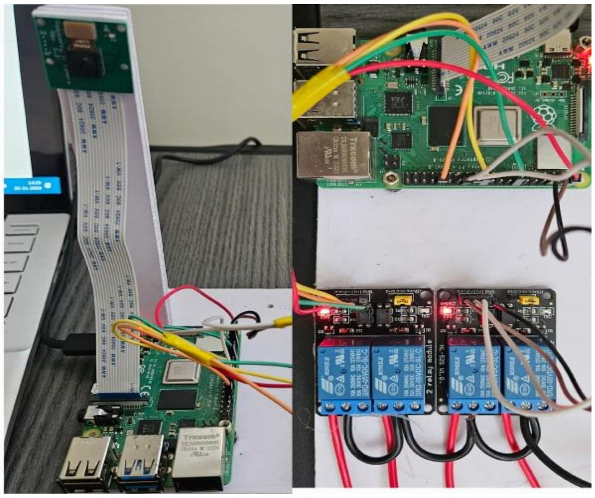

Fig. 4 Implementation of the proposed system

6.1. System Representation when it is in On and OFF Mode

Enhancement of timer, logging and integration of the system. timer automatically turns off the system after a certain period of inactivity. Logging the ON/OFF transitions to a file for monitoring. Integration of this logic to control physical devices like LEDs or motors through hardware

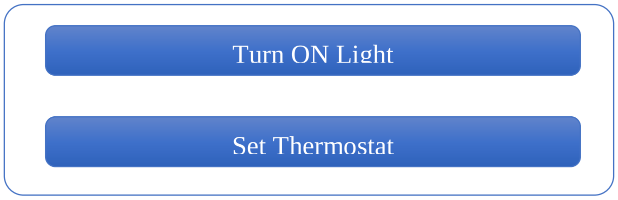

Fig. 5 System representation when it is in On and OFF Mode

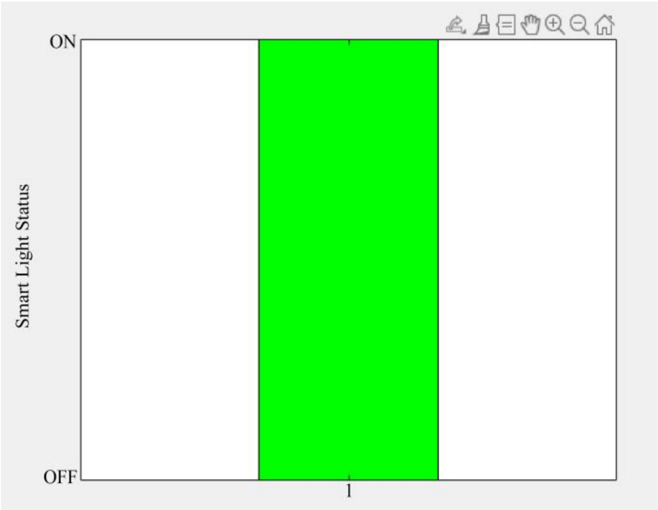

Fig. 6 Smart light status when it is ON

Fig. 7 Smart light status when it is OFF

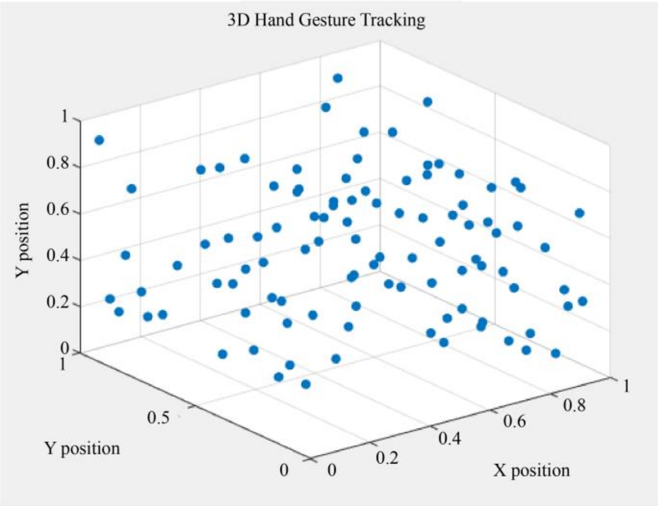

Fig. 8 3D hand gesture tracking

Using the 3D positions of the hand landmarks, the system can track how the hand moves in three-dimensional space, enabling recognition of gestures like swipes, pinches, or rotations that are mapped to specific actions.

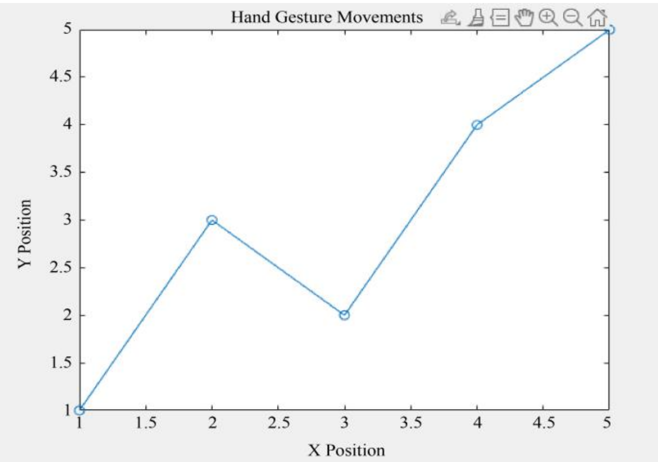

Fig. 9 Hand gesture movements

3D hand gesture tracking for home automation involves using computer vision to capture and interpret hand movements in three-dimensional space. Users can control smart devices such as lights, fans, and other home automation systems without physical contact by recognising specific gestures. The system relies on cameras, specialized algorithms, and machine learning models to detect hand movements and translate them into actions.

3D Gesture Recognition: After detecting the hand, the system uses the relative positions of the hand landmarks to estimate the hand's 3D position. In simple terms, while the camera provides a 2D image, algorithms infer the third dimension (depth) based on the distance between the landmarks or by using depth sensors.

The gesture recognition process involves analyzing the motion or configuration of the hand. If the hand moves rapidly in one direction (e.g., horizontally for left/right or vertically for up/down), the system interprets this as a swipe. Fist gestures: The hand is closed into a fist, and the system detects this through the proximity of the fingertips to the palm. Using the 3D positions of the hand landmarks, the system can track how the hand moves in three-dimensional space, enabling recognition of gestures like swipes, pinches, or rotations that are mapped to specific actions.

7. Conclusion & Future Scope

Radio waves, computer vision, machine learning, and embedded systems are the innovative technologies the Hand Gesture Controlled Electric Appliances system has incorporated to carry out different activities wirelessly and effortlessly. The hardware of Peach Lemon Soft is represented by a Raspberry Pi 4, which works with real-time video taken by a Pi camera. An open hand or a fist can be considered a hand gesture, and the role of OpenCV is to help locate and analyze these facial features. Such Command Control through a Relay Module connected with Raspberry Pi GPIOs can Control Electric Devices using Machine Learning Models that classify those gestures.

However, the system has problems such as Real-time Rendering and lighting changes. These problems have been solved as normalizing the colors is used to combat light variance while optimization methods relieve the need for long delays in the recognition of gestures. The relay module is used to turn on appliances through a switch. In the future, implementing speech commands into the system would be beneficial, as well as using more than one finger for gesture space in fingers. In addition, adding more advanced devices into the control from the level of a smart house can further improve this system's usability and make the home automation process easier and more pleasant. In the future, machine learning techniques can be used to analyze various features of hand gestures.

References

[1] Rafiqul Zaman Khanand, and Noor Adnan Ibraheem, "Hand Gesture Recognition: A Literature Review," *International Journal of Artificial Intelligence & Applications*, vol. 3, no. 4, pp. 161-174, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[2] Noraini Mohamed, Mumtaz Begum Mustafa, and Nazean Jomhari, "A Review of the Hand Gesture Recognition System: Current Progress and Future Directions," *IEEE Access*, vol. 9, pp. 157422-157436, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] N. Kumaran, M. Sri Anurag, and M. Sampath, "Hand Gesture Recognition Using Transfer Learning Techniques," *Journal of Current Research in Engineering and Science*, vol. 4, no. 1, pp. 1-8, 2021. [Google Scholar] [Publisher Link]

[4] Tuan Linh Dang et al., "An Improved Hand Gesture Recognition System using Keypoints and Hand Bounding Boxes," *Array*, vol. 16, pp. 1-10, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Samy Bakheet, and Ayoub Al-Hamadi, "Robust Hand Gesture Recognition using Multiple Shape-Oriented Visual Cues," *EURASIP Journal on Image and Video Processing*, vol. 2021, pp. 1-18, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Bharti Kumari et al., "Hand Gesture Recognition," *International Journal of Novel Research and Development*, vol. 7, no. 5, pp. 1327-1330, 2022. [Publisher Link]

[7] Jindi Wang et al., "Hand Gesture Recognition for User-Defined Textual Inputs and Gestures," *Universal Access in the Information Society*, pp. 1-15, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Swetha Kotavenuka et al., "Hand Gesture Recognition," *International Journal for Research in Applied Science and Engineering Technology*, vol .11, no. 1, pp. 331-335, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ruben E. Nogales, and Marco E. Benalcazar, "Hand Gesture Recognition Using Automatic Feature Extraction and Deep Learning Algorithms with Memory," *Big Data and Cognitive Computing*, vol. 7, no. 2, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Emrah Gingir "*Hand Gesture Recognition System*," Master Thesis, The Graduate School of Natural and Applied Sciences of Middle East Technical University, pp. 1-78, 2010. [Google Scholar]

[11] Camillo Lugaresi et al., "Mediapipe: A Framework for Building Perception Pipelines," *arXiv*, pp. 1-9, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Matthew D. Zeiler, and Rob Fergus, "Visualizing and Understanding Convolutional Networks," *Conference Proceedings 13th European Conference Computer Vision -- ECCV 2014*, Zurich, Switzerland, pp. 818-833, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[13] Carl Edward Rasmussen, and Christopher K. I. Williams, *Gaussian Processes for Machine Learning*," MIT Press, 2006. [Google Scholar] [Publisher Link]

[14] Niharika Ganji, Shriya Gandreti, and T. Rama Krishnaiah, "Home Automation Using Voice and Gesture Control," *7th International Conference on Communication and Electronics Systems*, Coimbatore, India, pp. 394-400, 2022. [CrossRef] [Google Scholar] [Publisher Link]