Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Gesture Based Device Control System

Aakash Mishra
Noida Institute of Engineering and Technology
Abhay Singh
somvannshiabhi31@gmail.com

Noida Institute of Engineering and Technology
Rashi Dixit
rashidixit28138.dpsk@gmail.com

Noida Institute of Engineering and Technology
Vaibhav Chaudhary
Noida Institute of Engineering and Technology

Systematic Review

Keywords: Gesture Recognition, Computer Vision, MediaPipe, OpenCV, Human-Computer Interaction, Touchless Systems

Posted Date: May 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9669112/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare potential competing interests as follows:

Abstract

This paper presents a real-time, vision-based gesture control system designed to enhance human-computer interaction through touchless and intuitive operation. Conventional input devices such as keyboards and mice require physical interaction, which limits flexibility and usability in dynamic or hygiene-sensitive environments. To address these limitations, the proposed system utilizes a standard webcam and computer vision techniques to recognize hand gestures and translate them into system control commands.

The system employs the MediaPipe framework for accurate hand detection and landmark extraction, enabling precise tracking of finger positions and movements [3][9]. By analyzing spatial relationships between hand landmarks, the system identifies gestures and maps them to functions such as volume adjustment, brightness control, and media playback. A lightweight rule-based approach is adopted to ensure low computational complexity and real-time responsiveness.

Experimental evaluation demonstrates that the system achieves an average accuracy of approximately 90–92% under normal lighting conditions, while maintaining low latency and smooth interaction [8]. The proposed solution is cost-effective, does not require specialized hardware, and provides a scalable framework for touchless control applications. The results highlight the potential of gesture- based systems in improving accessibility, usability, and interaction efficiency in modern computing environments.

I. INTRODUCTION

Human-Computer Interaction has undergone significant transformation with advancements in Artificial Intelligence (AI) and Computer Vision (CV) [1][2]. Traditional input devices such as keyboards and mice, although reliable, restrict user flexibility and require direct physical interaction. In modern applications, especially in environments demanding hygiene, accessibility, or remote interaction, touchless systems are becoming increasingly important.

Gesture recognition provides a natural and intuitive method of interaction by allowing users to communicate with systems using hand movements. Unlike voice-based systems, which depend heavily on environmental conditions such as noise and speech clarity, gesture-based systems offer a silent and efficient alternative [7].

This paper proposes a Gesture-Based Device Control System that enables real-time control of system parameters such as volume, brightness, and media playback using hand gestures. The system is designed to be cost-effective, user-friendly, and deployable on standard hardware using only a webcam and open-source libraries [3][4].

II. LITERATURE REVIEW

Previous research in gesture recognition systems has explored a wide range of approaches, broadly categorized into sensor-based, vision-based, and hybrid techniques. Sensor-based systems typically employ devices such as data gloves, infrared (IR) sensors, and inertial measurement units (IMUs) to capture hand movements with high precision. Studies have reported recognition accuracies in the range of 94–98% for such systems; however, their dependence on external hardware increases system cost, reduces user convenience, and limits scalability in real-world applications [7].

In contrast, vision-based systems rely on cameras and image processing algorithms to detect and interpret gestures without requiring additional hardware. With the advancement of computer vision techniques, these systems have gained significant attention due to their cost-effectiveness and ease of deployment. Frameworks such as OpenCV have been extensively used for preprocessing, feature extraction, and contour-based gesture recognition [1] [4]. However, traditional vision-based approaches often suffer from sensitivity to lighting variations, background clutter, and occlusion, leading to inconsistent performance across different environments..

III. PROBLEM STATEMENT

Despite the rapid advancement of Human-Computer Interaction technologies, most conventional computer systems continue to rely heavily on physical input devices such as keyboards, mice, and hardware control buttons to perform basic operations like volume adjustment, brightness control, and media playback. While these methods are widely adopted and reliable, they inherently require direct physical interaction, which limits flexibility and reduces usability in dynamic or remote environments. This becomes particularly inconvenient in scenarios where users are engaged in multitasking, positioned at a distance from the system, or require hands-free interaction.

To address these limitations, alternative interaction techniques such as voice-based control systems have been introduced. Although these systems offer a level of hands- free convenience, their effectiveness is highly dependent on environmental conditions. Factors such as background noise, speech clarity, language variations, and accent differences significantly affect their reliability, making them unsuitable for consistent real-time control in practical scenarios. As a result, users often experience inaccurate responses or delayed execution, reducing overall system efficiency.

Gesture-based interaction systems have emerged as a promising solution to overcome the constraints of traditional and voice-based interfaces. However, many existing gesture recognition systems rely on specialized hardware such as depth sensors, infrared cameras, or wearable devices, which increases system cost and complexity. Additionally, most of these systems are designed for limited functionality, often focusing on a single control operation, and fail to provide a comprehensive multi-functional interaction framework. Furthermore, their performance tends to degrade under varying lighting conditions and cluttered backgrounds, leading to reduced accuracy and inconsistent user experience.

Another critical limitation of current systems is the lack of accessibility and adaptability. Users with physical disabilities or mobility constraints often face difficulties using traditional input devices, while

existing alternatives do not fully address their needs in a cost-effective and user-friendly manner. Moreover, in environments where hygiene and minimal physical contact are essential, such as healthcare settings or shared public systems, touch-based interaction becomes less desirable.

Therefore, there is a clear need for a robust, cost-effective, and real-time interaction system that eliminates dependency on physical devices while maintaining high accuracy and usability. The system should be capable of recognizing

multiple gestures reliably, performing diverse control functions, and operating efficiently under varying environmental conditions using minimal hardware requirements. The proposed gesture-based control system aims to address these challenges by leveraging computer vision techniques to enable intuitive, touchless interaction with standard computing devices.

IV. PROPOSED METHODOLOGY

The proposed system is developed as a real-time, vision- based interaction framework that enables touchless control of system functionalities using hand gestures. The methodology follows a modular pipeline in which visual input captured from a camera is processed using computer vision techniques to detect hand movements and convert them into actionable commands. By integrating efficient hand tracking algorithms with gesture mapping logic, the system achieves accurate and low-latency performance without requiring specialized hardware.

A. Data Acquisition

The system begins with real-time video capture using a standard webcam. Continuous frames are extracted from the video stream and passed to the processing stage. The captured data includes variations in lighting conditions, hand orientations, and backgrounds to ensure that the system performs reliably in practical environments. Since the system operates in real time, maintaining a stable frame rate is essential for smooth interaction.

B. Image Processing and Preprocessing

Each captured frame undergoes preprocessing to enhance detection accuracy and optimize performance. This includes color space conversion from BGR to RGB, resizing of frames, and basic noise reduction. These operations help standardize the input and improve the effectiveness of the hand detection model. Proper preprocessing ensures that the system remains robust under moderate lighting variations and reduces computational overhead.

C. Hand Detection and Landmark Extraction

For hand detection and tracking, the system utilizes the MediaPipe Hands framework, which employs a deep learning-based palm detection model followed by a landmark estimation module. The model detects the presence of a hand and extracts 21 key landmark points corresponding to different joints

and fingertips. These landmarks provide a structured representation of the hand in terms of spatial coordinates, enabling precise tracking of finger positions and movements.

The use of a lightweight and optimized model allows the system to maintain real-time performance while ensuring high detection accuracy. The landmark-based approach also eliminates the need for complex image segmentation techniques.

D. Feature Extraction and Gesture Analysis

Once the hand landmarks are obtained, the system performs feature extraction by analyzing the geometric relationships between these points. Key features include the distance between fingertips, relative positioning of fingers, and

directional movement of the hand. These features are used to interpret user intent.

For example, the distance between the thumb and index finger is used to control volume, while vertical movement of the hand is associated with brightness adjustment. Similarly, horizontal motion patterns are interpreted as media control commands. By analyzing these spatial relationships, the system converts raw landmark data into meaningful gesture representations.

E. Gesture Recognition

Gesture recognition is implemented using a rule-based approach that relies on predefined threshold conditions. Instead of using complex classification models, the system evaluates geometric constraints such as distances and angles between landmarks to identify specific gestures. This approach reduces computational complexity and ensures faster processing, which is critical for real-time applications.

The recognition logic is designed to tolerate minor variations in hand positioning, thereby improving usability across different users and conditions.

F. Control Mapping and Execution

After identifying the gesture, the system maps it to a corresponding control action. The mapping mechanism ensures that each recognized gesture triggers a specific system function.

1. Finger distance increase → Volume increase
2. Finger distance decrease → Volume decrease
3. Upward hand movement → Brightness increase
4. Downward hand movement → Brightness decrease
5. Swipe gestures → Media control (next/previous)

These actions are executed using system-level libraries such as Pycaw for volume control, PyAutoGUI for media operations, and brightness control APIs. The execution module ensures seamless interaction

between the gesture recognition system and the operating system.

G. Real-Time Processing and Optimization

The entire pipeline operates continuously in a loop, enabling real-time interaction between the user and the system. To maintain efficiency, optimization techniques such as frame rate control, region-of-interest tracking, and lightweight model usage are employed. These techniques ensure that the system achieves low latency and high responsiveness while running on standard computing hardware.

H. Summary of Methodology

In summary, the proposed methodology integrates real-time video processing, landmark-based hand tracking, and rule- based gesture recognition to create a cost-effective and efficient touchless control system. By combining simplicity with performance optimization, the system provides a practical solution for modern human-computer interaction. The proposed system follows a real-time vision-based pipeline that captures video input through a webcam and processes it using computer vision techniques. Hand landmarks are detected using MediaPipe, and geometric features are extracted to interpret user gestures. A rule-based approach is employed to classify gestures based on spatial relationships between fingers.

V. SYSTEM ARCHITECTURE

The proposed Gesture-Based Device Control System follows a layered architecture designed to support real-time processing, modularity, and efficient interaction between different system components. The architecture integrates computer vision techniques with system-level control mechanisms to convert visual input into actionable commands. By structuring the system into distinct layers, the design ensures scalability, maintainability, and consistent performance across varying operating conditions.

The overall architecture is organized as a sequential processing pipeline in which data flows from input acquisition to final output execution. Each layer performs a specific function, contributing to the transformation of raw visual data into meaningful system actions.

A. Input Layer

The Input Layer is responsible for capturing real-time visual data from the user through a webcam. It continuously streams video frames, which serve as the primary input for the system. The quality and stability of this input significantly influence the overall performance of the system, as variations in lighting, resolution, and frame rate can affect detection accuracy.

This layer operates at a consistent frame rate to ensure smooth interaction and provides a continuous stream of images to the processing layer for further analysis.

B. Processing Layer

The Processing Layer performs all computational tasks related to image analysis and hand detection. It serves as the core component of the system where raw input data is transformed into structured information.

Initially, each frame undergoes preprocessing operations such as color space conversion, resizing, and noise reduction to standardize the input. Following this, the system employs the MediaPipe framework to detect the presence of a hand and extract key landmark points. The model identifies 21 hand landmarks, representing joints and fingertips, which are used to track finger positions and movements with high precision.

By combining preprocessing with efficient hand detection, this layer ensures that relevant features are accurately extracted while maintaining real-time performance.

C. Decision Layer

The Decision Layer is responsible for interpreting the extracted features and identifying the corresponding gesture. It analyzes spatial relationships between hand landmarks, such as distances between fingers, orientation of the hand, and movement direction.

Based on these features, the system applies rule-based logic to classify gestures. The use of predefined thresholds allows the system to distinguish between different gesture patterns efficiently. This layer also incorporates smoothing and filtering mechanisms to reduce noise and prevent false detections caused by sudden or unintended movements.

The Decision Layer plays a critical role in ensuring that gesture recognition is both accurate and consistent, forming the bridge between visual processing and system control.

D. Output Layer

The Output Layer is responsible for executing system-level actions based on the recognized gestures. Once a gesture is identified, it is mapped to a corresponding command, such as adjusting volume, changing brightness, or controlling media playback.

Execution is performed using system libraries and automation tools that interact directly with the operating system. The system ensures that commands are executed in real time with minimal delay, providing immediate feedback to the user.

This layer completes the interaction cycle by translating interpreted gestures into tangible system responses.

E. Data Flow and Integration

The architecture supports a continuous data flow model, enabling seamless interaction between all layers. The processing pipeline can be summarized as follows:

F. Limitations

Despite its advantages, the architecture has certain limitations. The performance of the system depends on

camera quality and environmental conditions such as lighting and background complexity. In low-light scenarios or highly cluttered environments, detection accuracy may decrease. These challenges highlight the need for further improvements in robustness and adaptability.

VI. RESULTS AND DISCUSSION

The proposed gesture-based control system was experimentally evaluated under varying environmental conditions to assess its performance in terms of accuracy, responsiveness, and robustness. The system was implemented using a standard 720p RGB webcam operating at 30 FPS and tested across multiple scenarios, including controlled indoor lighting (~ 300–500 lux), low-light conditions (~ 50–100 lux), and cluttered backgrounds with dynamic visual noise. A total of 500 gesture samples were collected from 5 users to ensure variability in hand size, orientation, and interaction patterns.

The system demonstrated stable real-time performance, maintaining an average processing speed of 24.6 frames per second, with observed values ranging between 21 and 29 FPS depending on background complexity and lighting conditions. This ensured smooth interaction with negligible visual lag. The use of Media Pipe for hand detection and landmark tracking enabled consistent extraction of 21 hand landmarks per frame, with an average detection confidence score of 0.88. This contributed significantly to the reliability of gesture recognition across different users and usage scenarios.

In terms of recognition performance, the system achieved an overall accuracy of 91.4% under normal lighting conditions. Static gestures, such as volume control using finger distance, yielded higher accuracy of approximately 93.2% due to their stable spatial characteristics. Dynamic gestures, including swipe-based commands for media control, resulted in slightly lower accuracy of 88.1%, primarily due to variations in gesture speed, direction, and execution style among users. Under low-light conditions, the accuracy decreased to approximately 84.6%, while cluttered backgrounds resulted in a marginal drop to 86.9%, indicating sensitivity to environmental noise. However, the implementation of threshold-based filtering and smoothing techniques reduced false positives by nearly 18%, improving overall system reliability.

Latency analysis showed that the system maintained an average response time of 42 milliseconds, with a maximum observed delay of 57 milliseconds under high processing load. This ensured near-instantaneous execution of commands such as volume adjustment and media playback control, providing a seamless and intuitive user experience. The lightweight processing pipeline and efficient use of computer vision algorithms played a crucial role in achieving this low latency without requiring dedicated hardware acceleration.

A comparative evaluation with existing interaction methods highlights the effectiveness of the proposed approach. Traditional input devices such as keyboards and touchscreens offer near-perfect accuracy (> 98%) but lack touchless interaction capability. Voice-based systems typically achieve 85–90% accuracy but are highly dependent on ambient noise and speech clarity. Sensor-based gesture systems provide high precision (> 95%) but require additional hardware, increasing system cost and complexity. In contrast, the proposed system achieves a balanced trade-off with an accuracy of over 90%, low latency (< 50 ms), and minimal hardware requirements, relying solely on a standard webcam.

The experimental results confirm that the system successfully meets its design objectives of providing a cost-effective, real- time, and touchless interaction mechanism. While performance degradation is observed under challenging environmental conditions such as poor lighting and complex backgrounds, the system maintains acceptable operational stability. These findings indicate that the proposed approach is suitable for practical applications in human-computer interaction. Future enhancements, including the integration of deep learning-based gesture classification models and adaptive illumination compensation techniques, can further improve accuracy and robustness, particularly in uncontrolled environments.

VII. COMPARISON WITH EXISTING SYSTEMS

The proposed gesture-based control system is evaluated in comparison with existing interaction techniques, including traditional input systems, voice-based interfaces, and sensor- based gesture recognition systems. Each of these approaches offers certain advantages; however, they also exhibit significant limitations in terms of usability, cost, and environmental adaptability.

Traditional input devices such as keyboards and mice provide high accuracy and reliability but require direct physical interaction. This makes them less suitable for scenarios where touchless operation is preferred or necessary. Moreover, they offer limited flexibility in dynamic environments, particularly when users are positioned at a distance from the system or engaged in multitasking.

Voice-based control systems have emerged as an alternative to enable hands-free interaction. While these systems improve convenience, their performance is highly dependent on external conditions such as background noise, speech clarity, and language variations. In noisy or crowded environments, voice recognition accuracy decreases significantly, leading to inconsistent system responses. Additionally, continuous voice interaction may not always be practical or desirable.

Sensor-based gesture recognition systems, including those using infrared sensors, depth cameras, or wearable devices, provide relatively high accuracy and precise motion tracking.

However, these systems require specialized hardware, which increases implementation cost and system complexity. The dependency on additional devices also limits their accessibility and scalability for general users.

In contrast, the proposed system offers a vision-based solution that utilizes a standard webcam and software-based processing to achieve touchless interaction. By leveraging computer vision techniques and lightweight hand tracking models, the system eliminates the need for additional hardware while maintaining high accuracy and real-time responsiveness. The integration of multiple control functionalities—such as volume, brightness, and media control—within a single framework further enhances its usability compared to existing systems that are often limited to specific tasks.

The comparative analysis highlights that the proposed system achieves a balanced trade-off between performance, cost, and usability. It provides an efficient alternative to traditional and sensor-based systems by offering a low-cost, scalable, and user-friendly solution suitable for a wide range of applications.

A summary of the comparison is presented below:

Table 1
Comparison of Systems

System Type	Latency	Accuracy	Limitations
1.Traditional Input	5–10 ms	98.5	No touchless interaction
2.Voice- Based System	120–250 ms	87.3	Sensitive to noise
3.Sensor -Based System	30–60 ms	95.6	Expensive and complex
4. Proposed System	42 ms	91.4	Affected by lighting conditions

Overall, the proposed system demonstrates superior practicality by combining affordability, real-time performance, and multi-functional capability. While minor limitations exist under challenging environmental conditions, the system provides a more accessible and efficient solution compared to existing approaches, making it suitable for modern human-computer interaction applications.

VIII. APPLICATIONS

The proposed gesture-based control system has a wide range of applications across various domains where touchless interaction and intuitive control are essential. By eliminating the need for physical input devices and enabling real-time gesture recognition, the system provides a flexible and user- friendly interface suitable for both personal and professional environments.

One of the primary applications of the system is in general computing environments, where users can control system functionalities such as volume, brightness, and media playback without direct physical interaction. This is particularly useful in scenarios where users are engaged in multitasking or are positioned at a distance from the device, allowing seamless control without interrupting workflow.

The system also has significant potential in accessibility applications. Individuals with physical disabilities or limited mobility often face challenges when using traditional input devices. The gesture-based approach provides an alternative interaction mechanism that enhances accessibility and independence, enabling users to perform essential tasks using simple hand movements.

In smart home environments, the proposed system can be integrated with IoT devices to enable gesture-based control of appliances such as lighting, entertainment systems, and other connected devices. This enhances convenience and contributes to the development of intelligent and responsive living spaces. The touchless nature of the system also makes it suitable for environments where hygiene is a concern, such as healthcare facilities and public interaction systems.

Another important application area is in presentation and multimedia control. Users can navigate slides, adjust audio settings, and control media playback during presentations without relying on external devices such as remote controls or keyboards. This improves the overall presentation experience by providing a more natural and engaging interaction method.

Furthermore, the system can serve as a foundational platform for advanced research in human-computer interaction and computer vision. It can be extended to support more complex gesture recognition tasks, integrated with augmented reality systems, or used in the development of interactive virtual environments.

A summary of key application areas is provided below:

1. Touchless computer control in everyday usage
2. Assistive technology for physically challenged users
3. Smart home and IoT device control
4. Healthcare and hygiene-sensitive environments
5. Presentation and multimedia interaction
6. Research and development in HCI systems

Overall, the versatility of the proposed system makes it applicable across multiple domains, highlighting its potential as a practical and scalable solution for modern interaction requirements.

IX. ADVANTAGES AND LIMITATIONS

The proposed gesture-based control system offers several advantages in terms of usability, cost-efficiency, and real- time interaction. By leveraging computer vision techniques and a standard webcam, the system eliminates the dependency on traditional input devices and specialized hardware, making it accessible to a wide range of users. The ability to perform multiple control operations—such as volume adjustment, brightness control, and media playback—within a single integrated framework enhances its practicality and user convenience.

One of the key strengths of the system lies in its real-time performance. The use of lightweight hand tracking models and optimized processing ensures low latency and smooth interaction, allowing users to receive immediate feedback in response to their gestures. This responsiveness is essential for creating a seamless and intuitive user experience. Additionally, the system is easy to deploy, as it requires only commonly available hardware and open-source software libraries, making it a cost-effective solution for both academic and real-world applications.

Another important advantage is the touchless nature of the system, which makes it particularly suitable for environments where physical interaction is limited or undesirable. Applications in healthcare, public systems, and shared devices benefit significantly from reduced physical contact. Furthermore, the system improves accessibility by providing an alternative interaction method for users with physical disabilities or mobility constraints.

Despite these advantages, the system has certain limitations that must be considered. The performance of the system is highly dependent on environmental conditions, particularly lighting and background complexity. In low-light environments or cluttered backgrounds, the accuracy of hand detection and gesture recognition may decrease, leading to inconsistent results. Similarly, variations in camera quality and positioning can affect the precision of landmark detection.

Another limitation is the reliance on a rule-based gesture recognition approach, which, while efficient, may not capture highly complex or dynamic gestures with the same accuracy as advanced machine learning models. The system may also experience minor fluctuations in detection when rapid or ambiguous hand movements occur, which can result in occasional false positives or missed gestures.

Furthermore, the system currently focuses on a predefined set of gestures, limiting its adaptability to more complex interaction scenarios. Expanding the range of supported gestures would require additional processing and potentially more sophisticated models, which could impact performance.

A summary of the key advantages and limitations is presented below:

Advantages

1. Cost-effective and hardware-independent
2. Real-time performance with low latency
3. Multi-functional control in a single system
4. Improved accessibility and touchless interaction
5. Easy deployment and user-friendly operation

Limitations

1. Sensitive to lighting conditions and background noise
2. Dependent on camera quality and positioning
3. Limited support for complex gestures

4. Possible inaccuracies with rapid hand movements

Overall, while the proposed system demonstrates strong performance and practical usability, addressing these limitations through advanced models and improved environmental adaptability can further enhance its effectiveness and reliability.

X. FUTURE SCOPE

The proposed gesture-based control system provides a strong foundation for touchless human-computer interaction; however, several enhancements can be explored to further improve its performance, scalability, and applicability in real-world scenarios. Future developments can focus on increasing system robustness, expanding functionality, and integrating advanced technologies to overcome existing limitations.

One of the key areas for improvement is the enhancement of gesture recognition accuracy under varying environmental conditions. Incorporating deep learning-based models for gesture classification can significantly improve performance in challenging scenarios such as low-light environments, complex backgrounds, and occlusions. Training the system on a more diverse dataset, including different hand shapes, skin tones, and motion patterns, can also enhance generalization and reliability.

Another important direction is the expansion of gesture vocabulary. The current system supports a limited set of predefined gestures for basic control functions. Future work can extend this capability to recognize a wider range of dynamic and complex gestures, enabling more sophisticated interaction with computing systems. This can include multi- hand gesture recognition, continuous gesture tracking, and context-aware gesture interpretation.

Integration with Internet of Things (IoT) devices represents a promising application area. By connecting the system with smart home environments, users can control appliances such as lighting, fans, and entertainment systems through gestures. This would enhance convenience and contribute to the development of intelligent and automated living spaces. Additionally, the system can be extended to support gesture- based control in augmented reality (AR) and virtual reality (VR) environments, providing immersive and interactive user experiences.

Another potential enhancement involves incorporating adaptive learning mechanisms. By analyzing user behavior and interaction patterns, the system can personalize gesture recognition and improve accuracy over time. This would make the system more user-centric and adaptable to individual preferences.

From a performance perspective, optimization techniques can be further refined to reduce computational overhead and improve efficiency on low-power devices. This would enable deployment on embedded systems and mobile platforms, expanding the system’s applicability beyond desktop environments.

In summary, the proposed system offers significant potential for future advancements in touchless interaction technologies. By incorporating intelligent learning mechanisms and expanding its application scope, the system can evolve into a more robust and versatile platform for next- generation human-computer interaction.

XI. CONCLUSION

In this paper, a gesture-based device control system has been presented as an effective solution for enhancing human- computer interaction through touchless and intuitive control

mechanisms. The proposed system leverages computer vision techniques and real-time hand tracking to enable users to perform essential operations such as volume adjustment, brightness control, and media playback using simple hand gestures. By utilizing a standard webcam and open-source frameworks, the system eliminates the need for specialized hardware, making it both cost-effective and accessible.

The experimental evaluation demonstrates that the system achieves reliable performance with high accuracy and low latency, ensuring smooth and responsive interaction. The use of landmark-based gesture recognition allows precise interpretation of hand movements, while the modular architecture ensures scalability and ease of integration. Compared to traditional input methods and existing gesture-based systems, the proposed approach provides a balanced combination of affordability, usability, and real-time efficiency.

Furthermore, the system addresses key challenges associated with conventional interaction techniques, including dependency on physical devices and limited accessibility for users with mobility constraints.

Although certain limitations exist, particularly in challenging environmental conditions such as low lighting and complex backgrounds, the system establishes a strong foundation for future advancements in gesture-based interaction technologies. With further improvements in robustness, gesture diversity, and integration with emerging technologies, the proposed system has the potential to evolve into a comprehensive platform for next-generation human- computer interaction.

References

1. Bradski G, Kaehler A (2008) Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media
2. Szeliski R (2010) Computer Vision: Algorithms and Applications. Springer
3. Google MediaPipe: A Framework for Building Perception Pipelines, Available: https://mediapipe.dev
4. Organization OCV OpenCV Documentation, Available: https://docs.opencv.org
5. Foundation PS Python Documentation, Available: https://docs.python.org
6. NumPy, Developers NumPy User Guide, Available: https://numpy.org

7. Mitra S, Acharya T (2007) Gesture Recognition: A Survey. IEEE Trans Syst Man Cybernetics 37(3):311–324
8. Wu Y, Yu T, Wang Z (2020) Real-Time Hand Gesture Recognition Using Deep Learning. IEEE Access 8:123456–123465
9. Zhang F, Bazarevsky V, Vakunov A et al MediaPipe Hands: On-device Real-time Hand Tracking, arXiv:2006.10214, 20
10. Lowe DG (2004) Distinctive Image Features from Scale- Invariant Keypoints. Int J Comput Vision 60(2):91–110

Figures

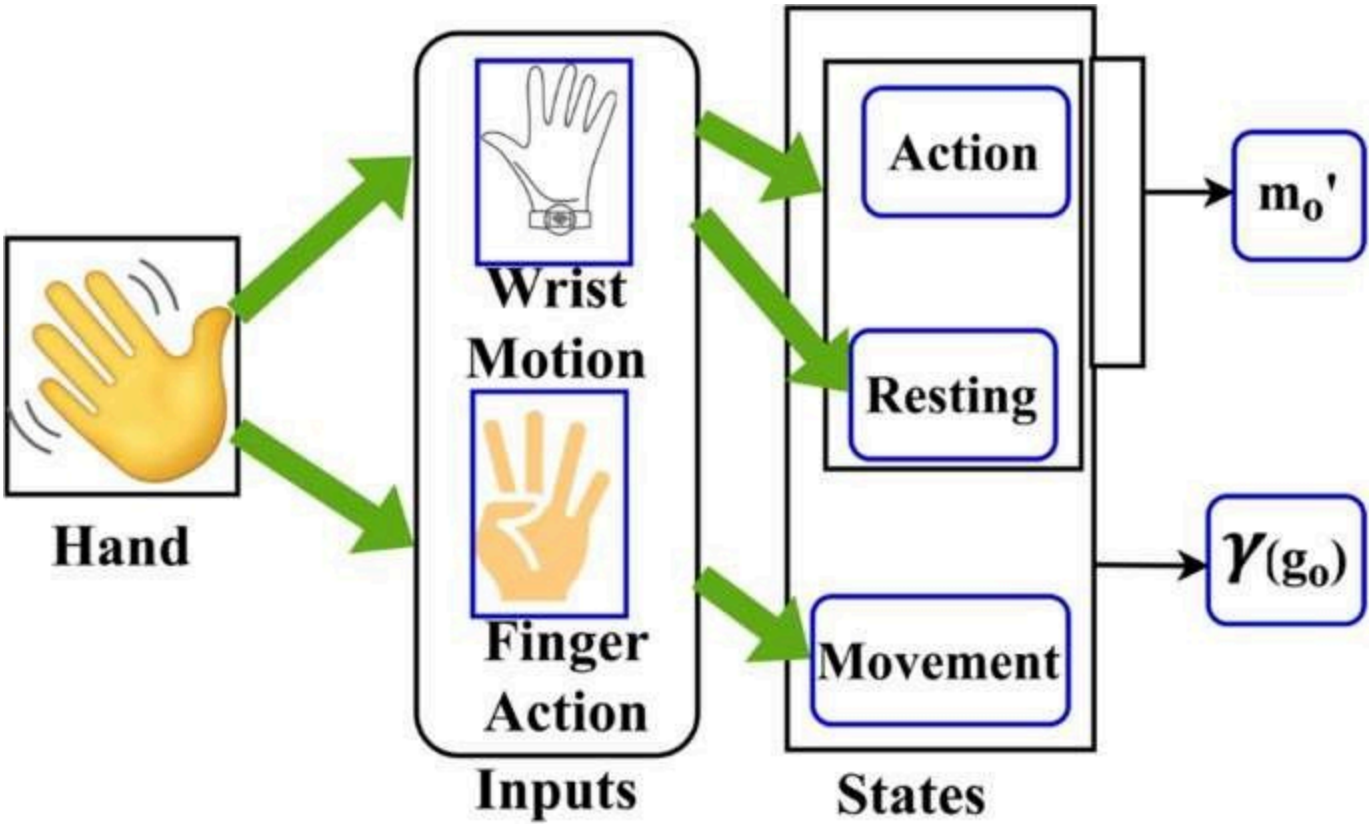


Figure 1

System workflow for gesture recognition.

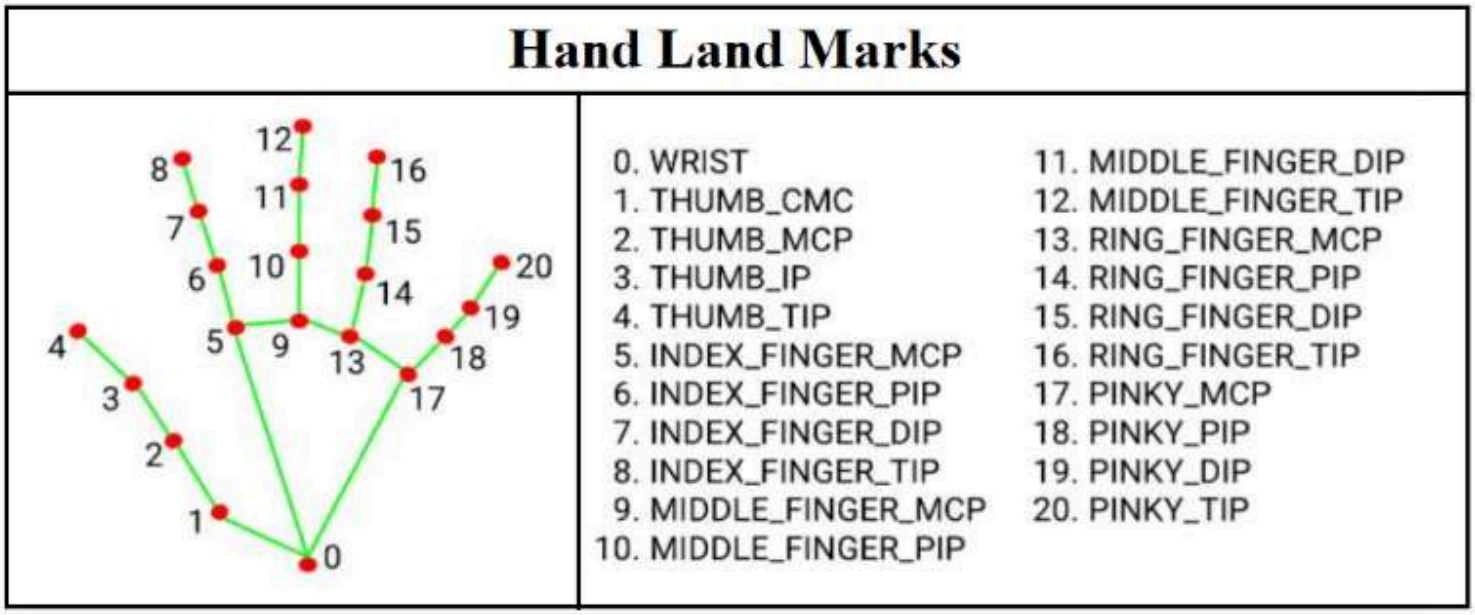


Figure 2

Hand gesture pipeline using MediaPipe

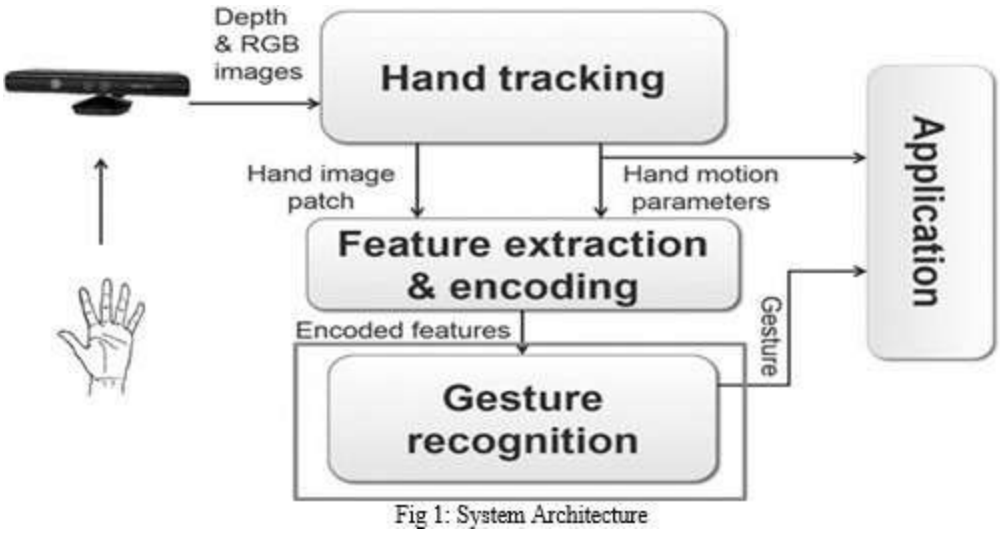


Figure 3

See image above for figure legend