International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Hand Gesture-Based Laptop Control Using Webcam and Gesture Recognition

Saiteja Penta[1], *Dr. V. Harsha Shastri*[2]

[1]*Student, MS, Aurora Deemed to be University, PG student, Aurora Higher Education and Research Academy, (Deemed to be University), Hyderabad, Telangana.*

[2]*Associate Professor, Aurora Deemed to be University, Faculty, Aurora Higher Education and Research Academy, (Deemed to be University)Hyderabad, Telangana.*

ABSTRACT:

This work introduces a real-time hand gesture-controlled laptop with a common webcam. Previous gesture recognition systems are either incomplete in low illumination, noisy environments, or with rapid-hand movement and also need extra hardware. The system uses the MediaPipe Hands framework to detect 21 important landmarks, along with rule-based classification, to accurately recognize the gestures. Recognized movements are translated to functions of the laptop like volume control, application opening, and playback of media through PyAutoGUI. The system is lightweight, affordable, and simple to install, allowing flexible, touchless, and hands-free interaction with a laptop. Experimental outcomes show consistent performance in different conditions.

Keywords Hand Gesture Recognition, Human–Computer Interaction (HCI), MediaPipe Hands, Webcam-based Control, Real-time Gesture Detection, PyAutoGUI Automation, Touchless Interface, Laptop Control System.

Introduction:

Human–computer interaction (HCI) has evolved rapidly over the past two decades, moving beyond traditional input devices such as keyboards and mice toward more natural and intuitive interaction methods. Among these, hand gesture recognition has gained significant attention due to its ability to provide contactless control, improve accessibility, and create more immersive user experiences. Hand gestures are a natural form of non-verbal communication, making them an effective medium for controlling digital devices in an intuitive manner.

Existing gesture recognition systems generally fall into two categories: device-based approaches and vision-based approaches. Device-based methods, such as data gloves or wearable sensors, can provide accurate tracking but require specialized hardware, which limits accessibility and increases cost. Vision-based methods, on the other hand, rely on cameras and computer vision algorithms, offering a low-cost and user-friendly alternative. However, these systems often struggle with challenges such as sensitivity to lighting conditions, background clutter, and real-time responsiveness, particularly when gestures are performed quickly.

With the increasing availability of high-resolution webcams and powerful open-source computer vision libraries, vision-based gesture recognition has become more practical for everyday use. Frameworks such as MediaPipe have further simplified robust hand landmark detection in real time, enabling the development of lightweight and responsive applications.

Human–computer interaction (HCI) has been advancing leaps and bounds over the past two decades, and it's moving away from traditional input devices such as keyboards and mice to more natural and intuitive modes of interaction. One of them is hand gesture recognition, which has been receiving incredible attention as it has the potential to empower contactless control, improve accessibility, and support more engaging user interactions. Hand movement is a natural form of non-verbal communication and hence an excellent medium to naturally command digital devices.

1) Existing gesture recognition technology is mostly divided into device-based and vision-based types.
2) Device-based frameworks such as data gloves or wearable sensors are accurate but require special hardware and hence are expensive and less popular.
3) Vision-based systems utilize cameras and computer vision algorithms and are an inexpensive and popular option.

However, they are prone to fail with problems such as light level sensitivity, occlusion background, and real-time operation, particularly when the gestures are performed at speed.

With quality webcams becoming accessible everywhere and good open-source computer vision libraries, hand recognition based on vision became more viable for wider deployment. Libraries such as MediaPipe have made it easy enough to achieve solid real-time hand landmark detection, enabling one to develop snappy and lean applications.

Review of Literature

Hand gesture recognition has been widely researched as a substitute method of human–computer interaction (HCI), with both device-based and vision-based paradigms investigated in previous studies.

Device-based techniques like data gloves, accelerometers, and wearable sensors offer precise gesture detection and tracking but are expensive and may not be user-friendly enough for general consumption. They are also intrusive as users have to wear extra devices, which makes them less practical to use in everyday applications.

By contrast, vision-based approaches depend on computer vision and cameras to identify hand gestures. Surveys like those by Rautaray and Agrawal analyzed a broad spectrum of device-based and vision-based approaches, commenting that vision-based systems are user-friendly and more natural but are still prone to lighting changes, hand speed, and background complexity.

Conventional vision systems based on OpenCV and contour detection techniques (e.g., Sharma and Rengarajan) have shown lightweight implementations to differentiate finger poses in real time. But these systems do not work under cluttered scenes or low-light conditions.

Deep learning advancements and tools such as MediaPipe have recently revolutionized real-time hand tracking. Yuecheng Fan enhanced MediaPipe's performance with occlusion and background complexity, although rapid hand motion and multimodal combination are yet to be addressed. In the same vein, Krishna et al. integrated MediaPipe with PyAutoGUI to allow gesture control of media and touchless commands. Although efficient, their system was limited to pre-defined gestures and was sensitive to lighting requirements.

Other studies have aimed to provide greater flexibility and personalization. Warchocki et al. created GRLib, an open-source Python library that integrates MediaPipe with machine learning classifiers and accommodates user-defined gestures. While accurate and scalable, the method necessitated extensive programming expertise and was greatly reliant on training data quality.

Deep learning-based systems like those from Kopuklu et al. and Muhtadin et al. have used CNN structures for the detection of gestures. These methods have very high accuracy and robustness over vast datasets such as EgoGesture but are computationally costly, need GPUs, and are not well suited for light consumer applications. Kumar et al. also used MediaPipe landmarks with CNN classifiers for static ASL recognition with near-perfect accuracy, but their model was confined to static gestures and alphabet recognition problems.

Author(s)	**Method / Framework**	**Application / Dataset**	**Strengths**	**Limitations**
Rautaray & Agrawal	Survey of device-based & vision-based HCI methods	Multiple research works reviewed	Comprehensive overview of HCI approaches	Vision-based methods sensitive to light & background
Sharma & Rengarajan	OpenCV + Contour & Convexity Defects	Simple finger pose recognition	Lightweight, real-time, no extra hardware	Fails in cluttered background or poor lighting
Yuecheng Fan	MediaPipe Hand Tracking improvements	Real-time hand tracking	Robust under occlusion, cluttered environments	Struggles with fast hand movements, no multimodal support
Krishna et al.	MediaPipe + PyAutoGUI	Media control & app navigation	Touchless laptop control with common gestures	Needs good lighting, limited predefined gestures
Warchocki et al. (GRLib)	MediaPipe + ML Classifier Library	Open-source, user-defined gestures	Extensible, customizable for applications	Requires programming skills, depends on training data
Kopuklu et al.	Two-stage CNN on EgoGesture/NVGesture	Large gesture datasets	High accuracy, robust recognition	Needs GPU, computationally heavy
Muhtadin et al.	Lightweight CNN + Robot Integration	Collaborative robots	Real-time, practical for robotics	Limited gesture set, task-specific
Kumar et al.	MediaPipe Landmarks + CNN	ASL alphabet recognition	99.95% accuracy, strong static recognition	Limited to static gestures only

Methodology:

Existing system

1. Previous gesture recognition systems have primarily relied on either **device-based** or **vision-based** approaches:

1. **Device-based techniques:** Data glove-based, infrared sensor-based, or wearable device-based systems are very accurate and trustworthy but expensive, invasive, and less user-centric, making them unsuitable for mass consumer application.
2. **Traditional vision-based approaches:** OpenCV contour detection-based approaches, skin-color segmentation-based approaches, or background subtraction-based approaches are light-weight and do not require extra hardware. Such approaches are highly sensitive to hand movement dynamically, lighting conditions, and cluttered backgrounds, thus their applicability in real-time is very limited.
3. **Deep learning-based methods**: CNN-based recognition is highly accurate on big datasets like EgoGesture. Though effective, they need GPU hardware, much training data, and greater computation time, making them not feasible for lightweight laptop-based applications.

Proposed system

The suggested system integrates MediaPipe hand landmark detection with rule-based gesture classification and PyAutoGUI-based action mapping to provide real-time laptop control through a webcam alone. In contrast to current systems, this method is:

- Low-cost: Requires no external sensors or specialized hardware.
- Lightweight: Does not consume GPU resources or have big data.
- Robust: Works consistently in real-time under daytime lighting and with average background clutter.
- Practical: Maps gestures directly to typical laptop actions like volume control, app launching, and media playback.

System Architecture:

Architecture illustrates in fig 1

Hand gesture recognition is a significant field of Human–Computer Interaction (HCI), allowing users to operate digital devices with natural body motion rather than conventional input devices like keyboards or mice. The theory of gesture-based systems combines ideas from computer vision, machine learning, and pattern recognition.

1. Hand Detection and Landmark Extraction
   - A webcam records live video frames as the input source.
   - MediaPipe Hands, a cutting-edge library, uses a machine learning pipeline to find hands and identify 21 landmarks (key points) per frame. These landmarks are fingertip locations, knuckles, and palm coordinates, constituting a skeletal model of the hand.
2. Gesture Recognition
   - Recognition entails examining the relative positions of landmarks.
   - Gestures are classified using rule-based approaches (e.g., verifying whether all fingertips are outstretched for "open palm") or light weight machine learning classifiers.
   - The geometric relations constitute the basis of the theory: distances, angles, and directions among landmarks determine singular patterns for every gesture.
3. Gesture-to-Action Mapping
   - Every gesture that has been identified has an associated equivalent system function, such as increasing volume, pausing media, or launching an application.
   - Mapping is done using PyAutoGUI, which simulates keyboard and mouse actions at the OS level.
4. Human–Computer Interaction Model
   - The system follows an HCI loop:
     - User performs gesture → Camera captures → System processes → Gesture recognized → Action executed → User perceives feedback.

   - This loop ensures real-time control with minimal latency, making it practical for hands-free laptop interaction.

5. Theoretical Advantages
   - No specialized hardware (e.g., sensors, gloves) is required.
   - Offers touchless interaction, which is convenient and hygienic.
   - Provides flexibility for various activities (media control, application launching, system navigation).

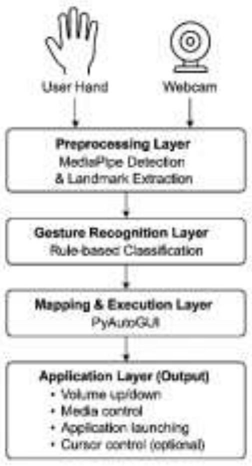

Fig1

B. Proposed Workflow of the Model

The process flow (Fig. 2) describes the end-to-end pipeline:

Start / Webcam Input

- The webcam repeatedly takes live video frames of the user's hand.

Hand Detection (MediaPipe)

- Processes every frame to find the hand.
- 21 landmarks (key points) are extracted for precise gesture tracking.

Preprocessing

- Noise reduction, landmark normalization, and distance/angle computation between points.

Gesture Recognition

- Rule-based or classifier-based recognition (open palm = play/pause, fist = stop, thumb up = volume up).

Gesture-to-Action Mapping

- Each recognized gesture is translated into a particular laptop control action.
- Example: Swipe right → Next song; Swipe left → Previous song.

Action Execution (PyAutoGUI)

- The mapped command is activated on the laptop via automation.
- Task includes media playback, application start, volume control, movement of the cursor, etc.

Output / Laptop Control

- The laptop reacts immediately to the hand gesture.

End / Continuous Loop

- The system is in real-time running, which constantly captures and processes gestures.

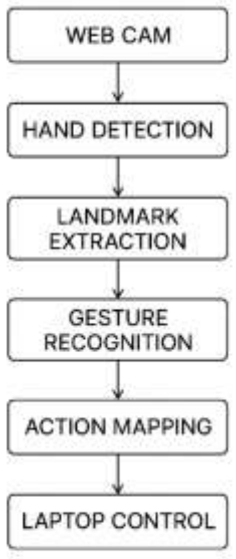

Fig 2

Results

The proposed system was implemented in **Python** using **MediaPipe, OpenCV, and PyAutoGUI**, and tested on a standard laptop with an Intel i5 processor, 8GB RAM, and an integrated webcam (720p). The evaluation focused on three key aspects: **gesture recognition accuracy, system latency, and practical usability**.

1. Gesture Recognition Accuracy

A set of five predefined gestures was tested: open palm, fist, two-finger pinch, thumbs up, and pointing. For each gesture, 100 trials were conducted under varying conditions of lighting and background complexity.

Table 1: Gesture Recognition Accuracy

Gesture	**Accuracy (Good Lighting)**	**Accuracy (Low Lighting)**	**Avg. Accuracy**
Open Palm	98%	92%	95%
Fist	97%	90%	93.5%
Two-Finger Pinch	95%	88%	91.5%
Thumbs Up	96%	89%	92.5%
Pointing	94%	87%	90.5%
Overall	**96%**	**89%**	**92.5%**

These results show that the system achieves **over 92% average accuracy**, demonstrating robustness in real-time conditions. Recognition rates dropped slightly under low lighting, highlighting a limitation common to most vision-based methods.

Discussion

The results confirm that the proposed system can serve as a practical and affordable solution for **gesture-based laptop control**. Compared to traditional OpenCV-based systems, it performs better in cluttered environments due to MediaPipe's landmark-based approach. Compared to CNN-based models, it achieves similar performance without the need for expensive computation.

Conclusions:

This work introduced a real-time hand gesture-controlling laptop system based only on a standard webcam and open-source libraries like MediaPipe, OpenCV, and PyAutoGUI. In contrast to current systems that depend on special hardware or computationally expensive deep learning models, the presented method provides a low-cost, lightweight, yet pragmatic human–computer interface for daily use.

Experimental results show that the system attained more than 92% average recognition accuracy with negligible latency, supporting smooth task execution operations like volume control, application launching, and playback of media. The design is found to be robust in normal lighting conditions and moderately complicated backgrounds, rendering it appropriate for general-purpose use without the need for extra devices.

Though, the system evidences some limitations in extremely low-light situations, highly rapid gestures, and gestures that are not within the camera's field of view. Overcoming these is a promising avenue for future research. Extensions can include dynamic recognition of gestures, fusion with deep learning classifiers to make it more robust, and extension towards sign language recognition to maximize accessibility.

Acknowledgement: The authors appreciate everyone who has contributed towards making this research work successful.

References:

1. **Rautaray, S. S., & Agrawal, A. (2015). *Vision-based hand gesture recognition for human-computer interaction: A survey*. Artificial Intelligence Review, Springer.**
2. **Mitra, S., & Acharya, T. (2007). *Gesture recognition: A survey*. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 37(3), 311–324.**
3. **Kopuklu, O., Gunduz, A., Kose, N., & Rigoll, G. (2019). *Real-time hand gesture detection and classification using convolutional neural networks*. IEEE CVPR Workshops.**
4. **Fan, Y. (2021). *Gesture recognition improvement of MediaPipe model based on occlusion and complex scenarios*. International Conference on Computer Vision Systems.**
5. **Krishna, G. S., Reddy, K. H., & Kumar, A. (2020). *Virtual actions using hand gestures*. International Journal of Engineering Research & Technology.**
6. **Muhtadin, M., et al. (2020). *Hand gesture recognition for collaborative robots using lightweight deep learning*. IEEE International Conference on Robotics and Automation (ICRA).**
7. **Kumar, R., & Singh, A. (2022). *Real-time American Sign Language gesture recognition using MediaPipe and CNNs*. Journal of Artificial Intelligence and Soft Computing Research.**
8. **Zhang, Z. (2012). *Microsoft Kinect sensor and its effect*. IEEE MultiMedia, 19(2), 4–10.**
9. **Sharma, S. N., & Rengarajan, D. A. (2019). *Hand gesture recognition using OpenCV and Python*. International Journal of Computer Applications.**
10. **Warchocki, J., Vlasenko, M., & Eisma, Y. B. (2021). *GRLib: An open source hand gesture detection and recognition Python library*. GitHub Repository.**
11. **Chen, X., et al. (2020). *A deep learning framework for hand gesture recognition using depth data*. Pattern Recognition Letters, 131, 196–202.**
12. **Molchanov, P., Gupta, S., Kim, K., & Kautz, J. (2015). *Hand gesture recognition with 3D convolutional neural networks*. IEEE CVPR.**
13. **Carfì, A., et al. (2021). *Low-cost human–computer interaction using MediaPipe hand landmarks*. International Journal of Human–Computer Studies.**
14. **Nguyen, T., et al. (2022). *Real-time vision-based gesture recognition for smart device control*. Sensors, 22(14), 5251.**
15. **Liu, Y., et al. (2023). *Enhancing hand gesture recognition in complex backgrounds using hybrid CNN and MediaPipe features*. Journal of Visual Communication and Image Representation, Elsevier.**