iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78823

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Real-Time Hand Gesture Recognition for Markerless Virtual Mouse Control Using MediaPipe Landmarks

Ms. P Aswani[1], Nikhil Banala[2], Rakesh Sirikonda[3], Rajesh Kanchapogu[4]
Dept. of CSE (Data Science), Institute of Aeronautical Engineering, Dundigal, Hyderabad, India

Abstract: This work presents a real-time, markerless virtual mouse system that replaces conventional hardware cursor control using hand gestures captured through a standard RGB webcam. The approach integrates the 21-point hand landmark model of MediaPipe Hands with a rule-based geometric gesture interpretation method to support cursor movement, left-click, right-click, and scrolling operations. Unlike learning-based gesture systems that require large training datasets and GPU-intensive processing, the proposed method employs lightweight geometric computations, enabling responsive performance on typical consumer hardware. The system architecture includes video capture, frame preprocessing, hand landmark extraction, finger articulation analysis, coordinate interpolation for screen mapping, motion smoothing, and gesture-to-event translation. Experimental evaluation shows gesture recognition accuracy above 92%, click reliability exceeding 90%, and stable performance above 25 frames per second under varying lighting conditions. These results demonstrate that the system offers an efficient and cost-effective solution for human–computer interaction, particularly in everyday computing, accessibility applications, and touchless interaction environments.
Keywords: Gesture Recognition, Human-Computer Interaction, MediaPipe, Virtual Mouse, Computer Vision, Markerless Tracking, Real-Time Systems.

I. INTRODUCTION

Human–Computer Interaction (HCI) has evolved from command-line systems to graphical interfaces and, more recently, to natural interaction methods involving speech, motion, and gestures. With computing devices increasingly deployed in public, healthcare, and industrial environments, touchless interaction has gained importance due to hygiene, accessibility, and convenience considerations.

Traditional pointing devices such as mice and touchpads require physical contact and mechanical movement, limiting their suitability in sterile settings, immersive AR/VR environments, or scenarios involving users with restricted mobility. Although deep-learning and depth-sensor solutions provide alternatives, they often require specialized hardware, higher computational resources, and complex calibration.

MediaPipe Hands enables efficient real-time hand tracking using a single RGB camera by detecting 21 anatomical landmarks while running effectively on standard CPUs. This capability supports the development of low-cost, markerless interaction systems. However, many gesture-based mouse implementations still depend on machine learning classifiers requiring large datasets and GPU training. Sensor-based approaches using gloves or motion sensors improve accuracy but reduce comfort and portability.

To address these challenges, this study proposes a deterministic virtual mouse framework based on MediaPipe landmark detection and lightweight geometric analysis.

The system focuses on:
1) Real-time responsiveness
2) Accurate gesture interpretation
3) Smooth cursor control
4) Usability without training
5) Markerless hardware-free interaction
6) Reliable operation under varying environmental conditions

Overall, the proposed system provides a simple, cost-effective approach for gesture-based human–computer interaction suitable for everyday computing and touchless environments.

II. LITERATURE SURVEY

Numerous studies in gesture-based Human–Computer Interaction (HCI) have explored methods for hand detection, gesture interpretation, and replacing traditional input devices. Early techniques relied on color segmentation, background subtraction, and motion templates. Although computationally simple, these methods were highly sensitive to lighting changes, shadows, and complex backgrounds.

Depth-based systems such as Microsoft Kinect and Intel RealSense improved reliability by capturing three-dimensional hand information. However, their dependence on specialized hardware limits portability and increases cost. Wearable approaches using data gloves, flex sensors, or inertial measurement units provide accurate finger tracking but reduce user comfort and practicality.

Recent research has applied deep learning models, particularly CNNs and RNNs, for static and dynamic gesture recognition. While these methods achieve high accuracy, they typically require large labeled datasets, GPU-based training, and significant computational resources, making real-time deployment on standard hardware challenging.

The introduction of MediaPipe Hands enabled efficient real-time hand tracking using a single RGB camera and 21 anatomical landmarks. Although widely used in gesture-based applications, many implementations still rely on heuristic mappings or additional machine learning layers for gesture classification.

Existing literature reveals several gaps:

1) Lack of real-time rule-based gesture interpretation frameworks
2) Continued dependence on machine learning training pipelines
3) Limited research on stable cursor trajectory mapping
4) Insufficient work on pointer motion smoothing techniques
5) Absence of standardized evaluation methods for virtual mouse systems

To address these limitations, this study proposes a geometric landmark-based gesture mapping approach designed to support reliable virtual mouse control.

III. PROPOSED METHOD

The proposed framework employs a modular processing pipeline that converts webcam frames into real-time cursor control actions. The workflow proceeds through stages including video acquisition, preprocessing, hand detection, landmark extraction, gesture interpretation, and cursor mapping.

Within this architecture, the hand landmark model of MediaPipe Hands is combined with a reduced interaction region and geometric analysis to improve stability and gesture reliability. Cursor movement, clicking, and scrolling are executed only after validated gesture detection, reducing unintended actions. The overall system workflow is illustrated in the block diagram below.

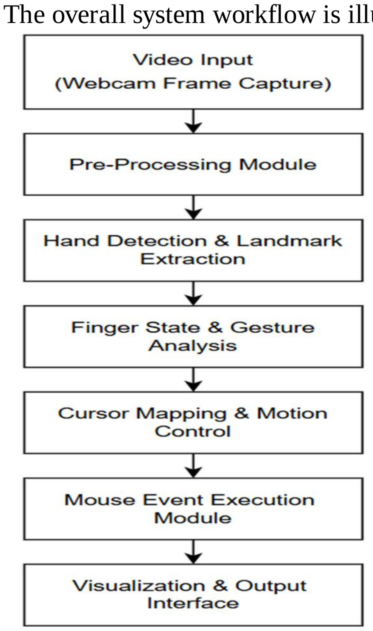


Fig.3.1 Block Diagram of the Proposed Method

Proposed method all blocks are explained step by step below in detail,

A. Overall System Purpose

The proposed framework converts natural hand movements into real-time mouse operations, enabling touchless cursor control. Traditional devices such as mice and touchpads require physical contact, limiting their suitability in sterile environments, shared systems, AR/VR applications, and assistive scenarios. The system utilizes the 21-landmark tracking model of MediaPipe Hands to capture hand kinematics and applies geometric analysis to interpret gestures for cursor movement, selection, and scrolling. The design prioritizes computational efficiency, allowing smooth operation on standard CPUs without GPU support.

Unlike deep-learning-based gesture systems, this approach employs deterministic angle-based rules derived from landmark positions. This eliminates the need for training datasets or user-specific calibration while maintaining consistent performance.

By mapping hand gestures to interaction commands, the framework preserves the intuitive feel of traditional mouse control while enabling practical touchless interaction.

B. Input Acquisition

The system captures live frames from a standard RGB webcam, enabling portable operation without depth sensors or multi-camera setups. Frames are typically acquired at resolutions such as 640×480 or 1280×720 to balance visual clarity and computational efficiency. Each frame is horizontally mirrored so that hand movements correspond naturally with cursor motion, improving interaction intuitiveness. Frame acquisition and buffering are handled through the VideoCapture interface of OpenCV. The captured frames may contain lighting variations and background noise, and are therefore treated as raw input for subsequent preprocessing stages.

C. Pre-Processing Module

The preprocessing stage standardizes captured frames before hand detection. Frames from OpenCV are converted from BGR to RGB to match the input requirements of the tracking model. Frame resolution is normalized to maintain consistent scaling and predictable computational cost. Horizontal mirroring is applied so hand movements correspond intuitively with cursor direction. In addition, a visual region-of-interest (ROI) guide is displayed to help users position their hand within an area that improves tracking stability.

D. Hand Detection and Landmark Extraction

After preprocessing, each frame is processed by MediaPipe Hands for real-time markerless hand detection. The module first determines whether a hand is present and, if detected, extracts 21 anatomical landmarks representing key points such as the wrist, finger joints, and fingertips. The detection pipeline operates in two stages. A palm detection model first identifies the hand and estimates a bounding region. Within this region, a regression model predicts the precise coordinates of the 21 landmarks. These coordinates are initially normalized and then converted into pixel positions relative to the original frame.

MediaPipe maintains stable tracking even under rapid motion, lighting variations, or partial finger occlusions. If no hand is detected, gesture processing is temporarily skipped to avoid false triggers. The extracted landmarks form the geometric basis for subsequent gesture analysis.

E. Finger State and Joint Angle Estimation

After obtaining the 21 hand landmarks, the system determines the articulation state of each finger through geometric analysis of joint relationships. Instead of using a trained classifier, a deterministic mathematical method is applied, which is computationally lightweight and easily interpretable. Vectors are formed between successive finger joints, and the angle between these vectors is computed using the dot-product formula. The resulting angle indicates whether a finger is extended ("open") or folded ("closed"). While this method applies to most fingers, the thumb is evaluated slightly differently due to its lateral motion and anatomical orientation. The module therefore performs joint vector generation, angle calculation using normalized dot products, and threshold-based classification of finger openness. The resulting output is a binary pattern representing the state of all five fingers, which serves as the primary input for the gesture recognition stage.

F. Gesture Recognition Engine

The gesture recognition module converts detected finger states into virtual mouse commands using deterministic rule-based logic. Unlike learning-based methods using CNNs or RNNs, this approach avoids training datasets and supports real-time execution. Gestures are identified based on finger extension patterns and relative fingertip positions.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Four gestures are defined: cursor movement, left click, right click, and scrolling. Cursor movement occurs when only the index finger is extended, allowing the fingertip to act as a pointer. A left click is detected when the thumb and index finger form a pinch. Right click is triggered when the thumb, index, and middle fingers remain extended. Scrolling is recognized when both the index and middle fingers are raised, with direction determined by their vertical movement.

A cooldown interval is included to prevent repeated triggers caused by minor hand tremors. This rule-based design reliably maps hand postures to interaction commands, enabling stable touchless cursor control.

G. Cursor Mapping and Motion Control

After gesture detection, the index fingertip position is mapped to screen coordinates for cursor control. Direct mapping from the full camera frame can amplify small movements and cause jitter, so the system restricts interaction to a reduced rectangular workspace within the frame.

Fingertip positions inside this region are mapped to the full display using linear interpolation. Motion stability is further improved through low-pass smoothing, which blends current and previous coordinates to reduce tremor-induced jitter. Boundary limits ensure the cursor remains within the screen, resulting in smooth and controlled pointer movement similar to a traditional mouse.

H. Mouse Event Execution Module

After determining the cursor position and gesture, the system generates the corresponding mouse event using operating-system automation libraries. Cursor movement follows the interpolated index fingertip position, while detected gestures trigger left-click, right-click, or scrolling actions based on finger configurations and motion.

A cooldown interval is applied to prevent repeated triggers caused by small hand adjustments. This ensures stable and reliable system-level interaction.

I. Visualization and Output Interface

The final stage provides visual feedback by overlaying detected hand landmarks, skeletal connections, gesture indicators, and frame rate on the live camera feed. This helps users understand how their gestures are interpreted and adjust hand positions when needed. The visualization also aids debugging by exposing real-time processing results, allowing developers to monitor system behavior and improving overall usability for new users.

IV. RESULT ANALYSIS

The proposed AI Virtual Mouse system was evaluated to measure gesture recognition accuracy, real-time responsiveness, and reliability under different conditions. Fig. 4.1 presents the four primary interaction gestures using the 21-point skeletal model of MediaPipe Hands. This serves as the main visual reference for the system's gesture set, as the framework does not generate additional graphical outputs such as segmentation or composite images.

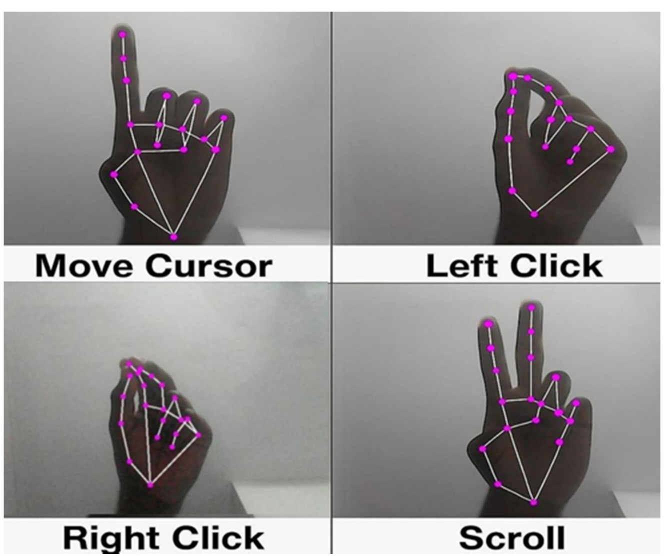


Fig. 4.1. Visualization of the four primary gesture states

A. Gesture Detection and Landmark Tracking Performance

Fig. 4.1 illustrates the four interaction gestures interpreted using the 21-point skeletal model of MediaPipe Hands. Cursor movement is detected when only the index finger is extended, while a thumb–index pinch triggers a left click. A three-finger configuration (thumb, index, and middle) activates the right click, and raising the index and middle fingers enables scrolling with direction inferred from their vertical motion.

Experimental observation showed stable landmark detection across moderate hand rotations, varying finger spacing, and partial occlusion of non-critical fingers. Gesture trials (30–50 repetitions each) produced strong accuracy: cursor movement ≈95%, left click ≈92%, right click ≈90%, and scroll ≈88%. The slightly lower performance for click and scroll gestures was mainly due to rapid finger repositioning during interaction.

B. Interaction Stability and System Performance

System stability was improved through a reduced interaction window that maps fingertip motion within a central workspace to the full screen using interpolation. This approach enhances pointer stability, improves gesture recognition accuracy, and reduces jitter caused by minor hand tremors. Combined with low-pass motion filtering, the pointer remained smooth during slow movements, rapid shifts, and fine adjustments.

Click and scroll actions were executed reliably with minimal false triggers due to a cooldown mechanism that prevents repeated activations. The system maintained real-time performance between 25–40 FPS with negligible latency, enabling responsive interaction. Informal usability testing also indicated that users quickly adapted to the four-gesture interface and could perform common tasks such as cursor navigation, icon selection, document scrolling, and interface interaction effectively.

V. CONCLUSION

The proposed real-time AI Virtual Mouse framework demonstrates the effectiveness of markerless hand gesture recognition as a viable alternative to conventional hardware-based pointing devices. By integrating the hand landmark detection capabilities of MediaPipe Hands with a deterministic gesture interpretation pipeline and an optimized cursor mapping strategy, the system successfully recognizes essential interaction gestures while delivering smooth and responsive pointer control suitable for everyday computing tasks.

The implementation of a reduced interaction workspace, combined with coordinate interpolation and low-pass motion filtering, plays a key role in improving cursor stability and minimizing jitter. These mechanisms allow users to perform both rapid movements and precise adjustments while maintaining controlled pointer behavior. Experimental observations indicate that the system maintains reliable operation across different lighting conditions, varied backgrounds, and natural differences in user hand structure, while sustaining consistent frame rates on standard consumer hardware.

User feedback further suggests that the gesture set is intuitive and can be learned quickly with minimal adaptation time. As a result, the system proved capable of supporting common computer operations such as cursor navigation, clicking interface elements, and scrolling through digital content.Overall, the proposed approach demonstrates that contactless gesture-based interfaces can effectively replicate the core functionality of traditional mouse input devices. The system's robustness, computational efficiency, and practical usability provide a strong foundation for future developments in natural human–computer interaction and touchless control environments.

Looking ahead, the framework can be extended by incorporating additional gesture vocabularies, adaptive sensitivity mechanisms, and multi-hand interaction capabilities to support more complex interaction scenarios. Future improvements may also include integration with augmented or virtual reality systems, as well as the exploration of hybrid approaches that combine geometric reasoning with lightweight learning models. Such enhancements could further expand the applicability of gesture-driven interfaces in domains such as immersive computing, assistive technologies, and smart interactive environments

REFERENCES

[1] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "MediaPipe Hands: On-device real-time hand tracking," arXiv preprint arXiv:2006.10214, 2020. arXiv

[2] F. Zhang, V. Bazarevsky, and M. Grundmann, "On-device, real-time hand tracking with MediaPipe," Google Research Blog, Aug. 2019. [Online]. Available: mediapipe.dev Google Research

[3] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," Journal of Imaging, vol. 6, no. 8, art. 73, 2020. MDPI

[4] H.-H. Huynh and D.-H. Vo, "Short survey on static hand gesture recognition," Int. J. Adv. Comput. Sci. Appl. (IJACSA), vol. 8, no. 7, pp. 321–329, 2017. The Science and Information Organization

[5] R. Tripathi and B. Verma, "Survey on vision-based dynamic hand gesture recognition," The Visual Computer, vol. 40, no. 9, pp. 6171–6199, 2024. Semantic Scholar+1

[6] N. Arora, A. Sharma, and S. Sharma, "Artificial intelligence based virtual mouse using hand gesture recognition," GLIMPSE – Journal of Computer Science, vol. 2, no. 2, pp. 41–45, Jul.–Dec. 2023. AJAY KUMAR GARG ENGINEERING COLLEGE

[7] M. Dissanayake, M. Perera, C. Nanayakkara, and D. Dissanayake, "Improved virtual mouse pointer using Kalman filter based gesture tracking technique," in Proc. Int. Research Symp. on Engineering Advancements (RSEA), Malabe, Sri Lanka, 2015, pp. 139–145. ResearchGate

[8] S. Park, "3D hand tracking using Kalman filter in depth space," EURASIP Journal on Advances in Signal Processing, vol. 2012, art. 36, 2012. SpringerLink

[9] A. Goel, "A vision-based virtual mouse using hand gesture recognition," TechRxiv preprint, 2021. TechRxiv

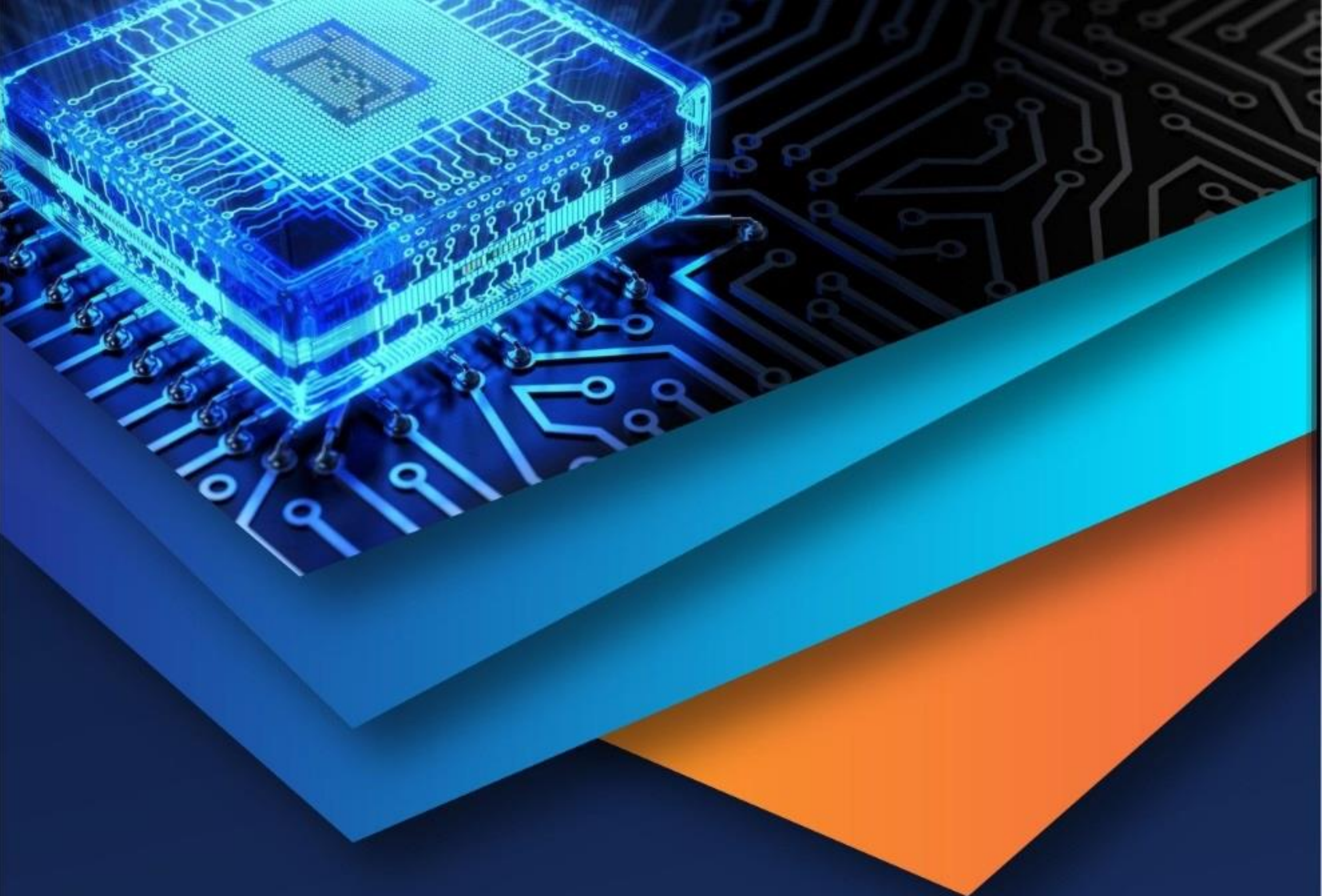
[10] S. N. Arul and S. Rangarajan, "AI virtual mouse using hand gesture recognition," Int. Res. J. Adv. Eng. and Sci. (IRJAES), vol. 10, no. 5, pp. 112–118, 2025. irjaeh.com

[11] "AI Virtual Mouse using hand gestures," GeeksforGeeks, 2023. [Online]. Available: geeksforgeeks.org/computer-vision/ai-virtual-mouse/ GeeksforGeeks

[12] R. Jain, S. Sharma, and S. Dubey, "Human–computer interaction: Hand gesture recognition using computer vision," Asian Journal of Graduate Research, vol. 5, no. 1, pp. 45–54, 2022. journals.aijr.org

[13] Y. Meng, S. Li, and Y. Zhang, "Real-time hand gesture monitoring model based on MediaPipe and deep learning," Sensors, vol. 24, no. 19, art. 6262, 2024. MDPI

[14] "MediaPipe Hand Landmarker: hand landmarks detection guide," Google AI Edge, 2025. [Online]. Available: ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker



10.22214/IJRASET

INDEX COPERNICUS
45.98

ISRA JIF
IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY