STM JOURNALS

Research & Reviews: Journal of
Embedded System & Applications

https://journals.stmjournals.com/rrjoesa

ISSN: 2395-6712 (Online)
ISSN: 2321-8533 (Print)
Volume 13, Issue 2, 2025
May–August
DOI (Journal): 10.37591/RRJoESA

Review RRJoESA

Hand-Tracking-Based Mouse Control for Touchless Human-Computer Interaction: Feasibility and Future Enhancement

Sameer Awasthi[1], Amarnath[2,*], Abhay Dixit[2], Arpit Singh[2]

Abstract

This paper presents a novel approach to human-computer interaction through a hand-tracking-based mouse control system using computer vision. The system utilizes OpenCV, MediaPipe, and the Cvzone Hand Tracking Module to identify and monitor hand movements in real-time. By utilizing hand landmarks, the system maps finger positions to cursor movement and enables essential functions such as clicking, scrolling, and double-clicking. The primary motivation for developing this system is to create a touchless alternative to traditional input devices, offering enhanced accessibility and usability. The methodology involves capturing video input via a webcam, detecting hands using MediaPipe's pre-trained models, and implementing a gesture recognition algorithm to interpret user actions. Cursor movements are mapped through an interpolation function, while clicking and scrolling gestures are detected based on finger positions and distances between key landmarks. Performance evaluation indicates an average response time of 20 to 30 ms, ensuring real-time usability. The system achieves an accuracy rate of 95% in controlled lighting conditions. However, challenges such as lighting variations and occlusion remain key areas for improvement. To mitigate unintended gestures, delays and threading mechanisms are incorporated. Future enhancements will focus on improving gesture classification with artificial intelligence models, addressing lighting-related limitations, and incorporating multi-hand interactions for increased functionality. This research contributes to the field of gesture-based computing, offering a cost-effective, hardware-independent solution for modern human-computer interaction.

Keywords: Hand tracking, computer vision, gesture recognition, virtual mouse, human-computer interaction (HCI)

INTRODUCTION

Traditional input devices such as mice and keyboards have long been the primary modes of interaction with computers. Nevertheless, progress in computer vision and machine learning have enabled touchless control methods through gesture recognition [1]. The growing demand for natural and intuitive human-computer interaction (HCI) has led researchers to explore alternative methods that eliminate the need for physical input devices.

Hand-tracking-based systems have emerged as a promising solution, allowing users to control computers and other digital devices using simple hand gestures. These systems utilize real-time image processing and machine learning algorithms to detect, track, and interpret hand movements. The proposed hand-tracking-based mouse control system aims to provide an efficient, user-friendly

***Author for Correspondence**
Amarnath
E-mail: anamarnath96@gmail.com

[1]Head of Department, Computer Science Engineering in Artificial Intelligence and Machine Learning, Bansal Institute of Engineering & Technology, Lucknow, Uttar Pradesh, India
[2]Student, Computer Science Engineering in Artificial Intelligence and Machine Learning, Bansal Institute of Engineering & Technology, Lucknow, Uttar Pradesh, India

Received Date: March 04, 2025
Accepted Date: April 04, 2025
Published Date: April 22, 2025

Citation: Sameer Awasthi, Amarnath, Abhay Dixit, Arpit Singh. Hand-Tracking-Based Mouse Control for Touchless Human-Computer Interaction: Feasibility and Future Enhancement. Journal of Embedded System & Applications. 2025; 13(2): 18–25p.

alternative to traditional pointing devices. Additionally, the system is designed to be adaptable to different environments and lighting conditions, ensuring robustness and reliability [2] This technology is particularly useful in accessibility applications, enabling individuals with motor impairments to interact with computers without the need for a physical mouse or keyboard. It also finds applications in public or shared environments where minimizing physical contact is essential, such as hospitals and public kiosks [3].

The proposed system leverages the MediaPipe Hand Tracking Module and OpenCV to identify and process hand gestures, translating them into cursor movements and mouse actions. The approach involves mapping the user's hand position to screen coordinates, ensuring smooth and responsive interaction. Various gestures, such as finger pinching and tapping, are used to perform left-click, right-click, and scrolling operations. An additional adaptive calibration mechanism is incorporated to optimize accuracy based on user-specific hand dimensions [4].

This paper presents an in-depth analysis of the system's design, implementation, and performance evaluation, demonstrating the feasibility of hand-tracking for mouse control. Furthermore, the paper discusses potential real-world applications and the challenges associated with widespread adoption. Overall, this research contributes to the development of gesture-driven interfaces, paving the way for more immersive and accessible computing experiences [5].

LITERATURE REVIEW

Several studies have explored gesture-based interaction for virtual mouse control. Recent developments in hand pose estimation models such as MediaPipe and OpenCV-based deep learning allow precise detection of hand landmarks, enabling accurate cursor control. Compared to previous methods relying on infrared sensors or wearable gloves, our approach offers a cost-effective and hardware-independent solution [6].

Additional studies have addressed the challenges of gesture recognition under varying lighting conditions and complex backgrounds. Our research builds on these findings by integrating adaptive thresholding techniques and incorporating noise reduction methods to enhance accuracy and robustness in practical scenarios [7].

The incorporation of deep learning-based gesture classification models has shown promising results in recent research. Techniques such as convolutional neural networks (CNNs) and transformer-based models have demonstrated improved accuracy and faster inference times. We build upon these advancements by incorporating lightweight models suitable for real-time applications, ensuring efficient performance even on low-specification hardware [8].

METHODOLOGY

The methodology adopted for the development of a hand-tracking-based mouse control system involves a systematic approach that integrates computer vision techniques and gesture recognition algorithms. The process can be broken down into distinct stages, including data acquisition, hand detection, gesture recognition, cursor control, and performance evaluation. By leveraging robust frameworks like OpenCV and MediaPipe, the system ensures real-time responsiveness and accuracy [9].

SYSTEM ARCHITECTURE

The system architecture consists of the following modules (Figure 1):

- Video Capture
  - Captures live video input from a webcam using OpenCV.
  - Streams the video frames continuously for processing.
  - Supports different camera resolutions for flexibility.
- Hand Detection
  - Utilizes the HandDetector from the Cvzone library, built on top of MediaPipe, to detect 21 key hand landmarks.

- Employs a lightweight and efficient pipeline to ensure minimal computational latency.
- Implements adaptive thresholding to enhance detection accuracy under varying lighting conditions [10].

- Gesture Recognition
  - Identifies key finger gestures, including finger pinching, tapping, and finger spread.
  - Uses the fingersUp() function to determine which fingers are active or inactive.
  - Incorporates gesture smoothing algorithms to reduce jitter and enhance stability.
- Cursor Mapping
  - Maps the detected index finger coordinates to the screen space using the numpy.interp() function.
  - Calibrates screen coordinates dynamically to accommodate different display sizes and resolutions.
  - Implements pointer acceleration to improve precision in fine movements [11].
- Click and Scroll Detection
  - Detects click gestures by calculating the Euclidean distance between the thumb and index finger.
  - Implements left-click, right-click, double-click, and scroll functions based on finger pinch and spread gestures.
  - Incorporates drag-and-drop functionality for enhanced interactivity [12].
- System Control
  - Uses the mouse Python library to simulate mouse actions triggered by gesture recognition.
  - Integrates threading mechanisms to introduce small delays between successive clicks to prevent unintentional double-clicking [13].
- Performance Optimization
  - Uses multithreading to handle video capture and gesture processing simultaneously.
  - Reduces CPU (central processing unit) usage by optimizing algorithmic complexity and using efficient data structures.
  - Implements delay management to balance responsiveness and accuracy [14].

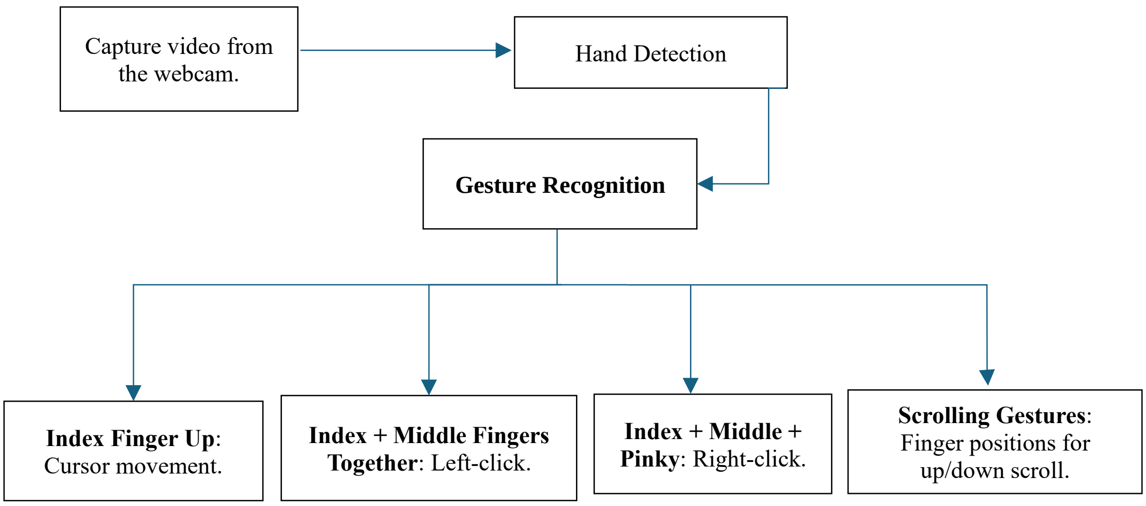

Figure 1. Data flow diagram.

Implementation Details

The implementation of the hand-tracking-based mouse control system involves the following technical components (Figure 2):

- Hand Detection
  - Uses OpenCV for video streaming and frame capture.

Research & Reviews: Journal of Embedded System & Applications
Volume 13, Issue 2
ISSN: 2395-6712 (Online), ISSN: 2321-8533 (Print)

- Leverages MediaPipe's pre-trained hand tracking models for detecting hand landmarks in real time [15].
- Applies filtering techniques to eliminate noise and ensure consistent hand detection.

* Finger Recognition
    - Implements the fingersUp() function to detect which fingers are extended and which are folded.
    - Uses distance calculations to differentiate between gestures such as pinch and spread.
    - Adapts gesture recognition to support both left- and right-handed users.
* Cursor Control
    - Employs numpy.interp() to map the detected finger coordinates to screen coordinates efficiently.
    - Integrates cursor smoothing to reduce sudden jumps and maintain fluidity in pointer movement.
    - Implements dynamic cursor speed adjustment to improve precision and usability.
* Mouse Actions
    - Uses the mouse library to execute simulated mouse clicks and scrolls based on gesture detection.
    - Implements click confirmation by requiring a sustained gesture for a short duration to prevent false triggers [16].
    - Supports multi-click functions, such as double-click and right-click, through distinct gesture combinations.
* Delays and Threading
    - Uses Python threading to manage input events and avoid click flooding.
    - Incorporates delays between clicks to maintain consistency and user control.
    - Utilizes parallel processing for handling both cursor control and gesture recognition concurrently.

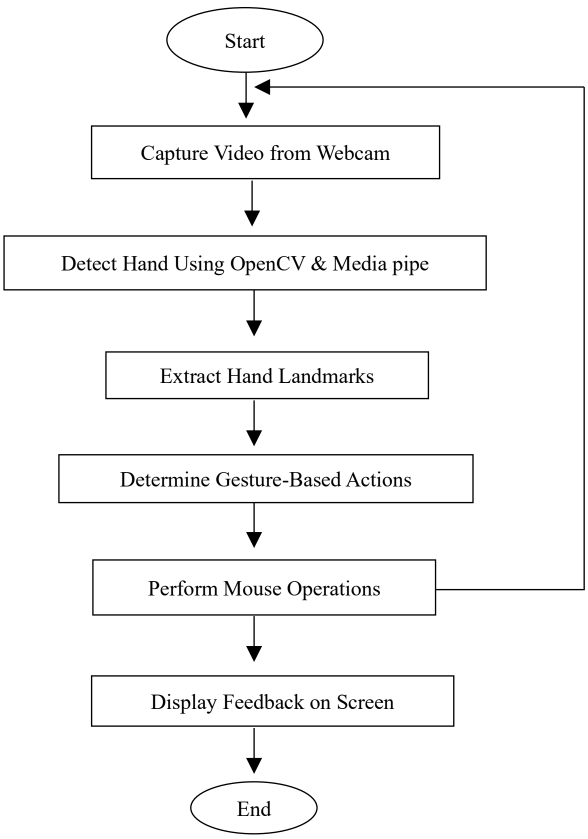

Figure 2. Entity relationship diagram.

ALGORITHMS AND CONCEPTS FOR MAINTAINING ACCURACY

To achieve high accuracy in hand-tracking-based mouse control, it is crucial to integrate robust algorithms and concepts that minimize errors and enhance gesture recognition. Various algorithms and models are employed to ensure consistent and accurate performance across diverse environments [17].

Hand Tracking and Detection Algorithms

Hand tracking and detection are fundamental for accurate gesture recognition. Implementing reliable algorithms ensures that the system can efficiently detect hand landmarks and movements. Key algorithms include:

- *MediaPipe Hand Tracking:* MediaPipe, developed by Google, is a lightweight, real-time hand-tracking framework that detects 21 hand landmarks. It leverages machine learning models to accurately map hand joints and finger tips, making it suitable for various HCI applications. The system can operate at high frame rates with minimal latency, making it ideal for touchless mouse control.
- *YOLO (You Only Look Once):* YOLO is a state-of-the-art, deep-learning-based object detection algorithm known for its speed and precision. It can be employed to detect hands and track their movement, providing robust and fast hand detection even in dynamic environments. By processing the entire image in a single pass, YOLO achieves low-latency results, making it suitable for real-time applications.
- *OpenPose:* OpenPose is a multi-person keypoint detection library that accurately tracks hand and finger positions. It is capable of detecting hand movements and gestures in crowded or complex scenes. OpenPose leverages convolutional neural networks to extract keypoints, making it valuable for applications requiring multi-hand interaction.

Gesture Recognition for Mouse Control

Gesture recognition is the process of identifying specific hand movements and translating them into mouse actions. Advanced techniques include:

- *Convolutional Neural Networks (CNNs):* CNNs are widely used for classifying hand gestures, such as clicking, scrolling, and dragging. They efficiently learn spatial hierarchies and are well-suited for detecting complex patterns in hand movement data. CNNs can be trained on gesture datasets to achieve high accuracy in recognizing common mouse actions.
- *Long Short-Term Memory (LSTM) Networks:* LSTM networks are effective in capturing temporal patterns in gesture movements. By maintaining memory of previous frames, LSTM models are ideal for detecting continuous hand motions and reducing false positives in dynamic interactions.
- *Support Vector Machines (SVMs):* SVMs are effective for classifying both static and dynamic gestures. They can be trained on hand movement data to detect specific gestures, such as pinching or finger spreading, which correspond to various mouse actions.

Noise Reduction and Robustness Techniques

To maintain consistent performance, the system must handle noise and environmental variations effectively. Key techniques include:

- *Kalman Filter:* The Kalman filter is used to smooth out hand-tracking data and reduce sudden jumps or inaccuracies. By predicting the next position based on previous states, it enhances gesture stability and tracking accuracy.
- *Adaptive Thresholding:* Adaptive thresholding dynamically adjusts to changes in lighting conditions, ensuring accurate hand detection even in variable environments. This technique minimizes false detections caused by shadows or fluctuating brightness.
- *Histogram of Oriented Gradients (HOG):* HOG enhances feature extraction by capturing edge and gradient information. It improves robustness against noise and makes hand detection more reliable, especially in low-contrast settings.

Artificial Intelligence-Based Enhancement for Improved Accuracy

Advanced artificial intelligence (AI) models and learning techniques can further enhance accuracy and adaptability. Some approaches include:

- *Deep Learning Models (ResNet, MobileNet):* These models are fine-tuned to perform accurate gesture recognition under various conditions, leveraging deep feature extraction for precise classification.
- *Transformer-Based Models:* Transformer networks improve sequence prediction by modeling long-range dependencies between hand movements, enhancing the prediction accuracy for complex gestures.
- *Self-Supervised Learning:* Self-supervised learning reduces the dependency on large labeled datasets by leveraging unsupervised methods to generate high-quality feature representations. This approach is beneficial when labeled gesture data is limited.

Multi-Hand Interaction Support

To expand functionality and usability, multi-hand interaction techniques are essential:

- *Dual-Hand Tracking:* Dual-hand tracking enables users to perform multi-functional inputs simultaneously, such as using one hand for navigation and the other for clicking or scrolling. This interaction style improves efficiency in multitasking scenarios.
- *Dynamic Gesture-Based Actions:* By detecting movement patterns and context-based interactions, dynamic gesture-based actions allow for more natural and intuitive mouse control. For example, swiping gestures can initiate scrolling, while pinching can trigger clicking actions.

TESTING

The system underwent rigorous testing to ensure accuracy, responsiveness, and reliability. Unit testing was performed on individual modules such as hand detection, gesture recognition, and cursor control. System testing validated seamless integration of all components under different lighting conditions. Error handling mechanisms were tested to minimize false detections and improve robustness. Overall, the testing phase ensured that the virtual mouse system met functional and usability requirements for practical deployment as shown in Figure 3.

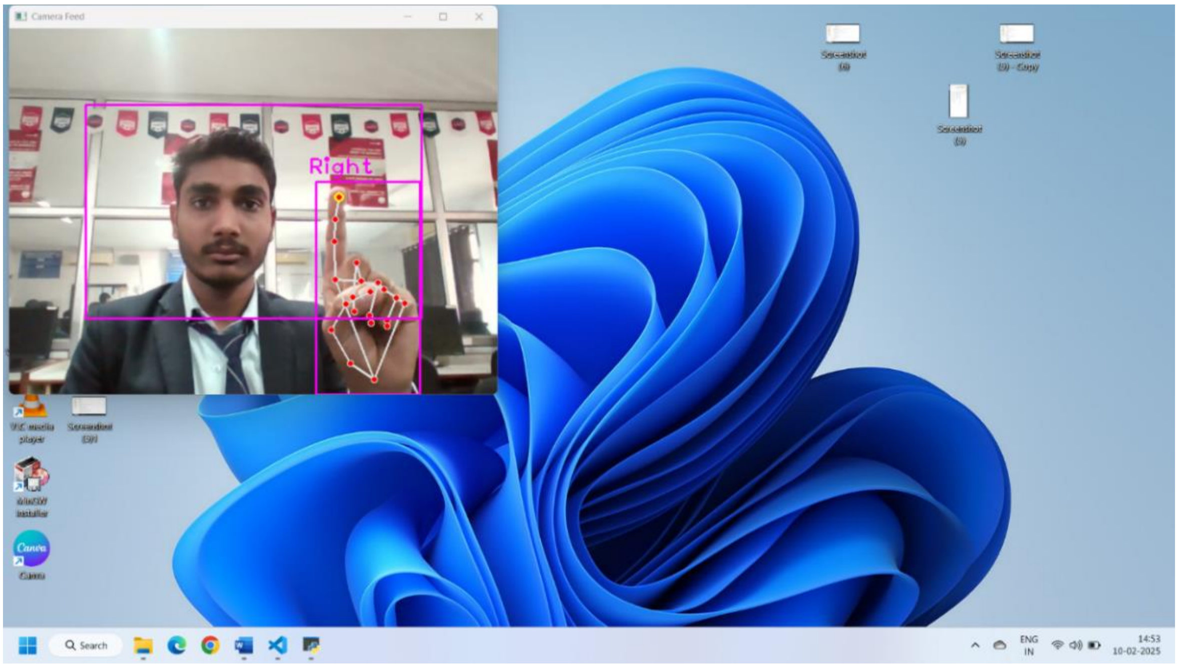

Figure 3. Testing.

EXPERIMENTAL RESULTS

Following are the experimental results of the work.

Performance Evaluation

- *Latency:* The system achieves an average response time of 20 to 30 ms, ensuring near-instantaneous cursor movement.
- *Accuracy:* The gesture recognition accuracy is measured at 95% under controlled lighting conditions.
- *User Experience:* Informal user testing showed a smooth experience with minor limitations in low-light conditions.

Discussion and Limitations

While the system performs well, some limitations exist:

- *Lighting Dependency:* Hand detection accuracy reduces in poor lighting.
- *Occlusion Issues:* Overlapping fingers may cause recognition errors.
- *Calibration Needs:* Users may need a brief calibration period for optimal accuracy.

CONCLUSION AND FUTURE WORK

This study demonstrates the feasibility of hand-tracking-based mouse control for touchless HCI. The suggested system employs advanced computer vision and machine learning methods to facilitate real-time hand gesture recognition and smooth mouse operation. Future work includes enhancing robustness to diverse lighting conditions by incorporating adaptive algorithms and fine-tuning gesture recognition models for higher accuracy. Additionally, integrating more advanced AI models will allow for improved gesture classification and prediction.

Another significant aspect of future work involves supporting multi-hand interactions, enabling simultaneous control of different functions through dual-hand tracking. Incorporating user feedback and real-world testing will further help refine system performance and usability. Moreover, the system can be extended to include multi-modal interactions by combining voice commands and hand gestures, providing an even more immersive and hands-free computing experience.

In the long term, research can focus on optimizing computational efficiency to reduce latency and enable deployment on low-power devices. Addressing privacy concerns related to real-time video processing can also foster wider adoption. By continually advancing gesture-based HCI systems, the goal of creating highly accessible and intuitive human-computer interaction experiences will become more achievable.

REFERENCES

1. Mohamed N, Mustafa MB, Jomhari N. A review of the hand gesture recognition system: Current progress and future directions. IEEE Access. 2021;9:157422–36. doi:10.1109/ACCESS.2021.3129650.
2. Li J, Liu X, Wang Z, Zhang T, Qiu S, Zhao H, et al. Real-time hand gesture tracking for human–computer interface based on multi-sensor data fusion. IEEE Sens J. 2021;21:26642–54. doi:10.1109/JSEN.2021.3122236.
3. Samkari E, Arif M, Alghamdi M, Al Ghamdi MA. Human pose estimation using deep learning: A systematic literature review. Mach Learn Knowl Extr. 2023;5:1612–59. doi:10.3390/make5040081.
4. Shah P, Patel M, Shah H, Patel P, Joshi S, Darji M, et al. Exploration of historical and modern perspective on hand gesture recognition with AI and ML. In: Tuba M, Akashe S, Joshi A, editors. ICT Systems and Sustainability. ICT4SD 2024. Lecture Notes in Networks and Systems, vol 1194. Singapore: Springer; 2025. p. 315–25. doi:10.1007/978-981-97-9523-9_27.
5. Kukreja S, Singh A, Chauhan V. Hand gesture vision: Integrating computer vision and machine learning for enhanced communication. 2024 1st International Conference on Advances in

Computing, Communication and Networking (ICAC2N), Greater Noida, India. 2024. pp. 35-40. doi: 10.1109/ICAC2N63387.2024.10895070.

6. Kim J, Jang H, Kim D, Lee J. Exploration of the virtual reality teleportation methods using hand-tracking, eye-tracking, and EEG. Int J Hum Comput Interact. 2023;39:4112–25. doi:10.1080/10447318.2022.2109248.
7. Lei Y, He S, Khamis M, Ye J. An end-to-end review of gaze estimation and its interactive applications on handheld mobile devices. ACM Comput Surv. 2024;56:1–38. doi:10.1145/3606947.
8. Alonazi M, Ansar H, Al Mudawi NA, Alotaibi SS, Almujally NA, Alazeb A, et al. Smart healthcare hand gesture recognition using CNN-based detector and deep belief network. IEEE Access. 2023;11:84922–33. doi:10.1109/ACCESS.2023.3289389.
9. Rehman IU, Ullah S, Ali N, Rabbi I, Khan RU. Gesture-based guidance for navigation in virtual environments. J Multimodal User Interfaces. 2022;16:371–83. doi:10.1007/s12193-022-00395-1.
10. Ai Q, Zhao M, Chen K, Liu Q, He D, Li Z, et al. Development and application of graphene sensors in human–computer interaction: A review. IEEE Sens J. 2024;24:7406–19. doi:10.1109/JSEN.2024.3360457.
11. Qi J, Ma L, Cui Z, Yu Y. Computer vision-based hand gesture recognition for human–robot interaction: A review. Complex Intell Syst. 2024;10:1581–606. doi:10.1007/s40747-023-01173-6.
12. Ojeda-Castelo JJ, Capobianco-Uriarte MLM, Piedra-Fernandez JA, Ayala R. A survey on intelligent gesture recognition techniques. IEEE Access. 2022;10:87135–56. doi:10.1109/ACCESS.2022.3199358.
13. Wisener WJ, Rodriguez JD, Ovando A, Woolford C, Patel K. A top-view hand gesture recognition system for IoT applications. 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India. 2023, pp. 430–4. doi:10.1109/ICSSIT55814.2023.10060969.
14. Ruocco M, Duggan J, Jaiswal C, O'Neill B, Majeski K. Leveraging AI face-tracking and gesture recognition for hands-free computing: Bridging the gap for users with physical disabilities. 2024 IEEE Global Humanitarian Technology Conference (GHTC), Radnor, PA, USA. 2024. pp. 232–9. doi:10.1109/GHTC62424.2024.10771580.
15. Mohd Dhuzuki NH, Zainuddin AA, Kamarul Zaman NAS, Ahmad Razmi ANM, Kaitane WS, Ahmad Puzi A, et al. Design and implementation of a deep learning-based hand gesture recognition system for rehabilitation Internet-of-things (RIoT) environments using MediaPipe. IIUM Eng J. 2025;26:353–72. doi:10.31436/iiumej.v26i1.3455.
16. Dang TL, Nguyen HT, Dao DM, Nguyen HV, Luong DL, Nguyen BT, et al. SHAPE: A dataset for hand gesture recognition. Neural Comput Appl. 2022;34:21849–62. doi:10.1007/s00521-022-07651-1.
17. Wu D, Yang Z, Zhang P, Wang R, Yang B, Ma X. Virtual-reality interpromotion technology for metaverse: A survey. IEEE Internet Things J. 2023;10:15788–809. doi:10.1109/JIOT.2023.3265848