A Vision-Based Virtual Mouse System Using Hand Gesture Recognition and AI

[1]**Dr.V.SHAVALI**
Associate professor,Department of ECE
Sri venkateswara institute of technology
Anantapur, India
v.shavali@gmail.com

[2]**MOUNIKA B**
Department of ECE
Sri venkateswara institute of technology
Anantapur, India
bondaledumounika@gmail.com

[3]**DIVYASREE A**
Department of ECE
Sri venkateswara institute of technology
Anantapur, India
appireddydivyasree3119@gmail.com

[4]**UMAKANTH REDDY** C
Department of ECE
Sri venkateswara institute of technology
Anantapur, India
chanapatiumakanthreddy@gmail.com

[5]**YASWANTH KUMAR B**
Department of ECE
Sri venkateswara institute of technology
Anantapur, India
billehaswanth@gmail.com

[6]**RAJESH Y**
Department of ECE
Sri venkateswara institute of technology
Anantapur, India
rajeshyamba188@gmail.com

[7]**Dr.VENKATESWARA REDDY VENNAPUSA**
professor,Department of ECE
Sri venkateswara institute of technology
Anantapur, India
vvreddy.ece@svitatp.ac.in

***Abstract*—Although touchscreen interfaces remain prohibitively expensive for desktop and workstation computers, they dominate modern handheld devices. This limitation can be addressed through an intuitive human–computer interaction approach, specifically a gesture-controlled virtual mouse operated via hand movements. This paper proposes a smart mouse system that captures and interprets hand gestures using a standard webcam. The system leverages OpenCV and advanced machine learning models to detect and analyze hand movements, enabling operations such as scrolling, left-clicking, right-clicking, and cursor navigation without the need for a physical mouse.**

***Index Terms*—Human–Computer Interaction, Hand Gesture Recognition, Object Tracking, OpenCV, MediaPipe, Palm Detection Model**

I. Introduction

The importance of Human-Computer Interaction (HCI) has increased significantly with the rapid advancement of computer technologies [**?**]. While touchscreens have become ubiquitous in mobile devices, they remain less practical and cost-effective for desktop systems [**?**]. In this context, computer vision-based virtual interaction tools provide a compelling alternative. Traditional input devices such as mice or trackpads are no longer essential, as virtual AI-driven mice can interpret hand movements to offer a smooth and intuitive method of interacting with computers [**?**].

This research proposes an AI-based system that tracks and interprets hand gestures using a standard webcam, along with MediaPipe, OpenCV, and advanced algorithms [**?**]. The system enables fundamental mouse operations—including cursor movement, left- and right-clicking, scrolling, dragging, volume and brightness control, and even file transfer via socket programming—through fingertip detection and gesture recognition [**?**]. By leveraging inexpensive hardware and computer vision techniques, this approach enhances both accessibility and the intuitiveness of human-computer interaction.

Beyond enabling natural user-computer communication, gesture recognition has diverse applications, including gaming, augmented reality, biomedical devices, and assistance for individuals with physical impairments [**?**]. To achieve accurate and real-time gesture detection, the system operates through four main stages: image preprocessing, region extraction, feature extraction, and feature matching.

Using hand gestures as an input method introduces unique challenges, particularly for users with physical limitations who may struggle with traditional devices such as mice or trackpads. Although gesture-based control provides a natural and expressive alternative, existing solutions are often either expensive or insufficiently precise for widespread adoption. To address these challenges, this study employs affordable webcams to develop a real-time hand motion recognition system. By utilizing hand gestures—a widely accepted and intuitive form of communication—the system eliminates the need for physical input devices. The ultimate objective is to create accessible and user-friendly desktop and laptop software that employs hand gesture recognition to facilitate interaction and comprehension in scientific and other applications.

II. Literature Review

A literature review systematically summarizes, analyzes, and synthesizes existing research related to a specific topic. It offers a critical examination of previously published studies, theoretical foundations, and methodological approaches that are pertinent to the research domain. Table 1 **??** presents a comprehensive comparison of various interaction modalities discussed in prior studies, highlighting their advantages, limitations, reliability, and application areas. In the context of

Citation	Modality	Advantages	Disadvantages	Reliability	Application
[1]	Speech Recognition	High accuracy in controlled environments; hands-free interaction	Limited in noisy environments; language-dependent	Moderate to High	Virtual assistants, accessibility tools
[2]	Gesture Recognition	Intuitive interaction; suitable for diverse use cases	Requires specialized hardware; computationally intensive	High in controlled settings	Gaming, robotics, medical systems
[3]	Vision-Based Systems	No physical contact required; supports complex interactions	Affected by lighting conditions; high computational demand	Moderate	Surveillance, AR/VR systems, automotive interfaces
[4]	Dataglove-Based Input	High accuracy for finger-specific gestures	Expensive; cumbersome to wear	High	Robotics, virtual reality, sign language recognition
[5]	Kinect Depth Cameras	Accurate depth sensing; works in low light	High cost; limited range	Moderate to High	Gesture recognition, skeleton tracking, accessibility systems

Fig. 1. Comparison of Modalities in Virtual AI Mouse Systems Using Hand Gestures .

virtual AI mouse systems and related fields, each modality represents a distinct technological approach or implementation strategy. This comparative analysis aids in identifying suitable techniques for gesture-based Human–Computer Interaction (HCI) by evaluating their feasibility, challenges, and practical relevance. Corresponding citations are included to support further investigation and validation of each modality. Figure

Fig. 2. Distribution of References by modality.

2 **??** presents the distribution of references across various journals and conference proceedings in the form of a pie chart. The analysis reveals a diverse range of sources, indicating that the cited studies adopted an interdisciplinary perspective.

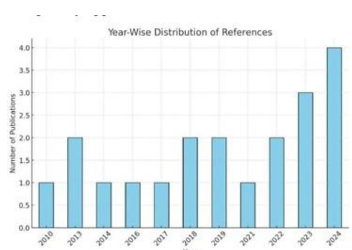

Fig. 3. Year Wise Publication Distribution

Figure 3 **??** illustrates the temporal trends in reference publications. In recent years, there has been a steady increase in the number of publications, reflecting the growing significance of gesture-based technologies. Notably, the years 2023 and 2024 exhibit pronounced peaks, highlighting a surge of interest and advancements in the field. References from earlier years, such as 2010 and 2013, indicate that the domain was still in its early stages at that time. Overall, these data provide insights into the diversity of research sources and the evolving interest in gesture-based systems over time.

III. Research Gap

Despite significant advancements in virtual AI mouse systems, several critical challenges remain. A primary issue is the lack of standardized evaluation metrics, which makes it difficult to effectively compare different approaches in terms of accuracy, latency, and usability across various modalities. Environmental robustness is another limitation, as motion-tracking and color-detection-based systems often struggle to operate reliably under crowded or varying lighting conditions. Affordability and accessibility also pose major barriers. High-cost technologies such as data gloves and Kinect sensors limit widespread adoption, highlighting the need for more cost-effective solutions that maintain comparable performance. The integration of multi-modal data remains a further challenge. Although some efforts have been made, seamlessly combining modalities like RGB-D data with deep learning for real-time applications is still difficult. Ergonomics and user adaptability are frequently overlooked; systems that fail to account for differences in hand size, shape, and motion can lead to discomfort and fatigue. Finally, real-time performance on edge devices is a significant concern, as high-accuracy systems typically demand substantial computational resources, making them impractical for devices with limited processing capabilities. Addressing these gaps can enable the development of virtual AI mouse systems that are more capable, reliable, and user-friendly, thereby promoting broader adoption and enhanced usability across diverse applications.

IV. Methodology

A. Hardware and Software Requirements

1) **Webcam:** A webcam serves as the Human-Computer Interaction (HCI) interface between the user and the system. It is essential for capturing images and tracking hand gestures. Higher camera resolutions enhance tracking accuracy and improve the overall user experience.
2) **Windows Operating System:** The Windows OS is recommended for development and testing due to its compatibility with the required libraries and tools.
3) **Python:** Python is the primary programming language for developing computer vision applications because of its simplicity, flexibility, and platform independence. It supports various libraries suitable for vision-based projects.
4) **OpenCV:** OpenCV (Open Source Computer Vision Library) is a powerful toolkit for image and video processing. It provides functionalities for tasks such as object detection, facial recognition, and advanced image manipulation. Its Python bindings make it easy to implement and suitable for rapid development. Installation can be done using:

```
pip install opencv-python
```

5) **MediaPipe:** Developed by Google, MediaPipe provides ready-to-use solutions for applications such as pose

estimation, face detection, and hand tracking. Its modular structure allows seamless integration into various applications, enabling real-time gesture recognition for virtual mouse control. Installation command:

```
pip install mediapipe
```

6) **AutoPy:** AutoPy is a cross-platform GUI automation library that allows control of the mouse and keyboard, including screen interaction through bitmap and color detection. It is used to simulate mouse movements and clicks. Installation command:

```
pip install autopy
```

B. Mathematical Modeling

1) **Euclidean Distance:** The Euclidean distance is used to identify hand gestures by measuring the distance between two hand landmarks:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the two points.

2) **Arctangent Function (arctan 2):** The angle at a point B formed by vectors $\overrightarrow{BA}$ and $\overrightarrow{BC}$ in 2D space can be calculated using:
Here, $\arctan 2(y, x)$ computes the angle of a vector (x, y) w.r.t the x-axis in radians, and θ is the difference b/w two angles.

C. System Architecture

1) **Input Frame:** The system first captures video frames from a connected camera.
2) **Hand Detection:** The system identifies the presence of hands within each frame. For real-time processing, the webcam continuously records the live video stream.

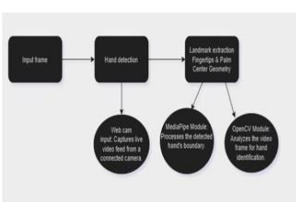

Fig. 4. Flowchart of Hand Detection and Landmark Extraction

D. Landmark Extraction

This stage involves detecting the hand and identifying key features such as the palm center and fingertips. The MediaPipe module processes the detected hand region and extracts essential landmarks, including fingertip positions. Meanwhile, the OpenCV module handles video frame analysis and supports hand detection.

The system combines image processing and predictive modeling to translate hand movements into mouse actions, such as cursor movement and clicks, enabling smooth and natural interaction.

E. Algorithm

The Virtual AI Mouse system follows these steps:

1) Capture video from the webcam.
2) Preprocess the video frames, including converting frames to the RGB color format.
3) Detect the hand using a Palm Detection Model.
4) Identify hand landmarks with the Hand Landmark Model.
5) Calculate distances between landmarks using the Euclidean distance formula and extract relevant features.
6) Recognize gestures by comparing the extracted features with predefined thresholds or criteria.
7) Map the identified gestures to corresponding mouse actions.
8) Execute system-level commands to simulate mouse movements and clicks on the screen.
9) Provide immediate visual or functional feedback to the user.

This method provides an intuitive alternative to traditional input devices by ensuring precise cursor control and reliable gesture recognition.

V. Result

A. Gesture Features

The primary gesture for controlling the cursor involves raising the index finger, often accompanied by the middle finger, as illustrated in Fig. **??**. This gesture serves as the main interaction mechanism with the system, allowing smooth and precise cursor movement across the screen. The second

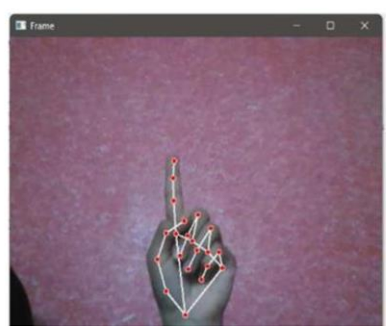

Fig. 5. Computer Window with a mouse controller for moving cursor around the computer

gesture, illustrated in Fig. **??**, performs the left-click operation. It involves raising the middle finger, extending the thumb, and bending the index finger. This gesture is essential for selecting or interacting with on-screen elements, mimicking the standard left-click action of a conventional mouse. The third gesture, shown in Fig. **??**, corresponds to the right-click function. It is executed by raising the index finger, extending the thumb, and bending the middle finger, enabling users to access context menus and additional options on the interface.

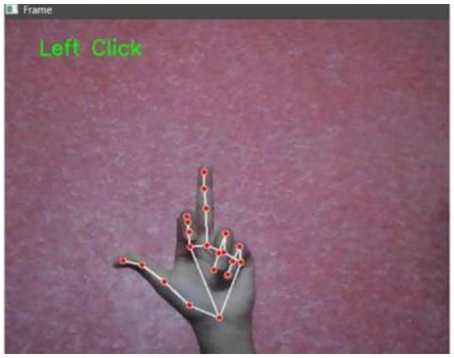

Fig. 6. Index finger bent and middle finger up to perform left click

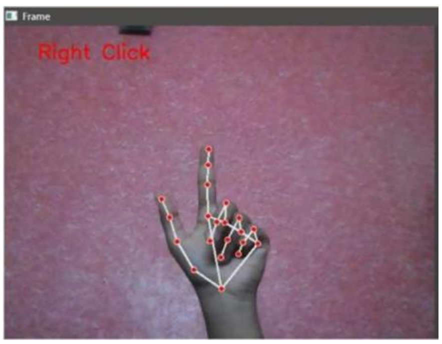

Fig. 7. Index finger up and middle finger bent to perform right click

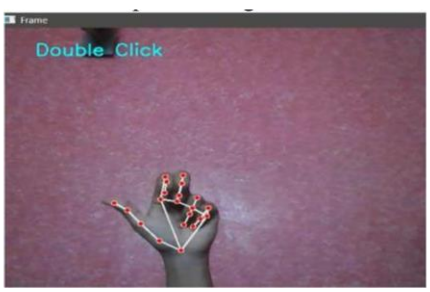

Fig. 8. Index and middle fingers closed to perform double click

B. Additional Gesture Features

The double-click gesture is performed by raising the index and middle fingers while keeping the thumb extended, as shown in Fig. **??**. This gesture replicates the standard double-click function of a conventional mouse and is useful for tasks such as opening files or launching applications.

The screenshot gesture is executed by closing all fingers, as illustrated in Fig. **??**. This provides an efficient method for capturing the screen without the need for extra hardware.

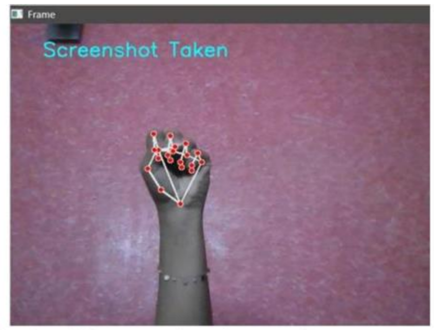

Fig. 9. All fingers closed to perform screenshot

Together, these gestures replace traditional input devices with intuitive hand motions, offering a simple and user-friendly way to control a virtual mouse.

C. Performance Analysis

The performance evaluation of the AI-powered virtual mouse system is summarized in Table **??**. The system achieves perfect accuracy (100%) for basic tasks such as cursor movement and taking screenshots, with no recorded failures.

Moderate accuracy (80%) is observed for the left-click and right-click gestures, each recording 20 failures. The double-click gesture performs slightly better, achieving 90% accuracy with 10 failures. Overall, the system records 450 successful actions and 90 failures, resulting in a total accuracy of approximately 83.33%.

These results demonstrate that the system is highly reliable for simpler gestures, while more complex actions like clicks may require further optimization to improve recognition accuracy.

Hand gesture	Task	Success	Failure	Accuracy
Index Finger up(Index and Middle Finger up)	Moving Cursor	100	0	100
Index finger bent and middle finger up	Left Click	80	20	80
Index finger up and middle finger bent	Right Click	80	20	80
Index and middle fingers closed	Double Click	90	10	90
All fingers closed	Screenshot	100	0	100
Result		450	50	450

Fig. 10. Performance Evaluation Table

Figure **??** illustrates the performance of the AI-based mouse gestures using a bar chart. The green bars represent the number of successful gesture recognitions for each task, while the red bars indicate the number of failures. Tasks such as "Move Cursor" and "Screenshot" achieved a perfect success rate of 100%, whereas "Left Click" and "Right Click" each recorded 20 failures, highlighting areas that could benefit from further improvement.

VI. Conclusion

The proposed virtual mouse system, which replaces traditional hardware devices with camera-based hand gesture

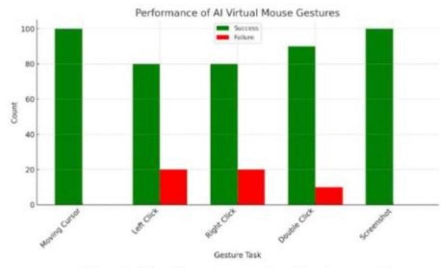

Fig. 11. Performance Analysis graph

recognition, represents a notable advancement in human–computer interaction. By accurately detecting hand movements, the system significantly enhances usability and accessibility, particularly for individuals with paraplegia. Experimental results demonstrate a precision of 98.6% in cursor movement as well as left-click, right-click, and double-click operations.

Although certain interactions such as dragging and scrolling are currently limited, ongoing progress in computer vision, machine learning, and related technologies is expected to address these shortcomings. Future improvements may involve the inclusion of a wider range of hand gestures, functional enhancements, and integration with emerging technologies such as augmented reality (AR) and virtual reality (VR). These advancements would allow the system to be deployed across multiple platforms, including smartphones and smart televisions, enabling a seamless and intuitive user experience. With continued refinement and expanded functionality, hand gesture-based virtual mouse systems have the potential to become versatile tools that redefine human–computer interaction and seamlessly integrate into everyday digital environments.

References

[1] L. Bretzner, I. Laptev, and T. Lindeberg, "Recognition of hand gestures using multi-scale colour features, hierarchical representations, and particle filtering techniques," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 423–428.

[2] V. Koles and V. Kruglinskiy, "A vision-driven approach to hand gesture recognition for human–computer interaction," *Journal of Visual Languages & Computing*, vol. 20, no. 4, pp. 204–210, Aug. 2009.

[3] J. C. G. da Silva, R. A. de Oliveira, and C. S. Yoneyama, "Applications of hand gesture recognition in human–computer interaction systems," in *Handbook of Human–Computer Interaction*, M. Soares et al., Eds. Cham: Springer, 2019, pp. 637–664.

[4] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision Using the OpenCV Library*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2008.

[5] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, Apr. 2017.

[6] Google Research, "MediaPipe: An open-source framework for constructing multimodal perception pipelines," [Online]. Available: https://mediapipe.dev/. [Accessed: Feb. 2026].

[7] X. Zhang, Q. Wang, and D. Zhang, "Real-time recognition of hand gestures through finger segmentation techniques," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 456–460.

[8] O. D. Lara and M. A. Labrador, "A comprehensive survey of hand gesture recognition methods for human–machine interaction," *ACM Computing Surveys*, vol. 45, no. 3, Art. 38, Jun. 2013.

[9] X. Huang and Y. Li, "Real-time hand tracking and gesture classification using convolutional neural networks," *Neurocomputing*, vol. 429, pp. 318–328, 2021.

[10] P. Viola and M. J. Jones, "Rapid object detection through a boosted cascade of simple feature classifiers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 511–518.

DR. V. SHAVALI Dr. V. Shavali, M.Tech, Ph.D, is an accomplished academician in the field of Electronics and Communication Engineering. He holds both M.Tech and Ph.D degrees from JNTUA, Anantapur, Andhra Pradesh. Currently, he serves as the Associate Professor, where he plays a key role in academic leadership and departmental develop ment. With over 18 years of rich teaching experience, Dr. Shavali has taught a wide range of core and advanced subjects in Electronics and Communica tion Engineering. His teaching approach emphasizes strong fundamentals, practical understanding, and continuous academic improvement, contributing significantly to student success and academic excellence. Dr. Shavali has also made notable contributions to research and academia, with several research publications to his credit. His areas of interest include emerging technologies in ECE, and he actively encourages research culture among students and faculty. Through his dedication to teaching, research, and administration, he continues to contribute meaningfully to the growth of the institution and the engineering community.

DIVYASREE A DIVYASREE A is currently pursuing a Bachelor of Technology degree in Electronics and Communication Engineering at Sri Venkateswara Institute of Technology. Her primary areas of interest include embedded systems and computer vision.

MOUNIKA B MOUNIKA B is currently pursuing a Bachelor of Technology degree in Electronics and Communication Engineering at Sri Venkateswara Institute of Technology. Her primary area of interest is embedded systems.

UMAKANTH REDDY C UMAKANTH REDDY C is currently pursuing a Bachelor of Technology degree in Electronics and Communication Engineering at Sri Venkateswara Institute of Technology. His primary area of interest is embedded systems.

YASWANTH KUMAR B YASWANTH KUMAR B is currently pursuing a Bachelor of Technology degree in Electronics and Communication Engineering at Sri Venkateswara Institute of Technology. His primary area of interest is embedded systems.

RAJESH Y RAJESH Y is currently pursuing a Bachelor of Technology degree in Electronics and Communication Engineering at Sri Venkateswara Institute of Technology. His primary area of interest is embedded systems.

Dr.VENKATESWARA REDDY VENNAPUSA Dr. Venkateswara Reddy Vennapusa (Member, ISTE) received the B.Tech. degree in Electronics and Communication Engineering from Madras Uni versity, Chennai, India, in 2003, and the M.Tech. degree in Digital Systems and Computer Electron ics from JNTU Anantapur, Andhra Pradesh, India, in 2010. He earned the Ph.D. degree in Applied Electronics from Gulbarga University, Kalaburagi, India, in 2025, where his doctoral research focused on Microstrip Antennas for Ultra-Wideband (UWB) applications. He is currently a Professor with the Department of Electronics and Communication Engineering at the Sri Venkateswara Institute of Technology (SVIT), Ananthapuramu, India. With a career spanning over 22 years, he has held significant leadership roles, including serving as a Head of Department. A dedicated mentor, he has successfully guided 53 student projects, comprising 32 undergraduate and 21 postgraduate theses. He has authored 21 research papers in reputed international journals and conferences. Dr. Reddy is a Life Member of the Indian Society for Technical Education (ISTE), a Member of the Institution of Electronics and Telecommunication Engineers (IETE), and a Member of the International Association of Engineers (IAENG). Notably, he is a licensed radio enthusiast, holding a Grade-II HAM Radio License (VU3TEE) issued by the Ministry of Communica tions, Government of India. His research interests include high-gain antenna design, RF systems, VLSI, and Mobile Adhoc Networks. He possesses technical expertise in VHDL programming and industrial automation systems, including PLC and SCADA. He is highly proficient in professional engineering and simulation tools, including HFSS, CADFEKO, IE3D, MATLAB, and LATEX