Virtual Mouse Using Hand Gesture Recognition and Computer Vision Using Python

S. Phani Varaprasad[1], Dr. R. Prasad Rao[2], V. Syamala[3], V. S. R. Chaitanya[4], B. Akhila[5]
P. Chandra Kiran[6]
[1]*Assistant Professor, Department of ECE, Avanthi Institute of Engineering and Technology*
[2]*Professor, Department of ECE, Avanthi Institute of Engineering and Technology*
[3,4,5,6]*Students, Department of ECE, Avanthi Institute of Engineering and Technology*

***Abstract*—The Virtual Mouse using Hand Gesture Recognition and Computer Vision is an advanced human-computer interaction system that enables users to control computer cursor operations through hand gestures without requiring physical contact with input devices. The proposed system utilizes computer vision and machine learning techniques through MediaPipe Hand Tracking and OpenCV libraries for real-time hand landmark detection and tracking.**

The movement of the cursor is achieved by mapping the index finger position to screen coordinates. Various gestures are employed to perform mouse operations such as left click, right click, scrolling, and drag-and-drop functionality. Pinching gestures between fingers are used for click operations, while finger movement patterns enable scrolling and brightness control.

The system is implemented using Python with an interactive Tkinter-based graphical user interface. The proposed approach provides an efficient, touchless, and user-friendly alternative to traditional mouse devices. It enhances accessibility, improves hygiene in shared environments, and contributes to the advancement of intelligent human-computer interaction systems.

***Index Terms*—Virtual Mouse, Hand Gesture Recognition, Computer Vision, Human-Computer Interaction(HCI), Media Pipe, OpenCV, Touch less Interface.**

I. INTRODUCTION

Human-computer interaction has evolved significantly with the advancement of artificial intelligence and computer vision technologies. Conventional input devices such as keyboards and mice require physical contact, which may not always be convenient or hygienic in modern applications. Gesture recognition systems provide a natural and touch-free method for interacting with computers.

The Virtual Mouse Using Hand Gesture Recognition and Computer Vision is designed to replace traditional mouse devices by enabling users to control cursor operations through hand gestures captured by a webcam. The system integrates computer vision techniques with machine learning algorithms to detect and track hand landmarks in real time.

The proposed system employs OpenCV for image processing, MediaPipe for accurate hand tracking, and PyAutoGUI for controlling mouse functions. This approach enables smooth cursor movement and efficient gesture-based operations such as clicking, scrolling, and dragging.

The system is particularly useful in healthcare environments, smart homes, gaming applications, and assistive technologies for physically challenged individuals. By eliminating the need for physical interaction, the system improves accessibility, hygiene, and user experience.

II. LITERATURE SURVEY

Hand gesture recognition has become an important research area in computer vision, artificial intelligence, and human-computer interaction. The increasing demand for touchless systems in healthcare, automation, and virtual environments has encouraged researchers to develop efficient gesture-based interaction systems.

Earlier gesture recognition systems depended on hardware devices such as data gloves, infrared sensors, accelerometers, and gyroscopes. Although

these systems provided accurate gesture tracking, they were expensive and required additional hardware components.

With the development of computer vision and deep learning techniques, camera-based gesture recognition systems gained popularity due to their affordability and scalability. Several methods have been proposed for gesture recognition and hand tracking:

- Template Matching: A conventional technique where predefined hand templates are matched with captured images. However, this method suffers from high computational complexity and limited real-time performance.
- Haar Cascade and HOG Features: These approaches were widely used for object and hand detection but had limitations in detecting dynamic hand gestures accurately.
- Convolutional Neural Networks (CNNs): Deep learning-based methods significantly improved gesture classification accuracy by learning complex hand features from large datasets.
- MediaPipe Framework: Developed by Google, MediaPipe provides efficient real-time hand tracking using 21 hand landmarks. It is lightweight, fast, and capable of operating using standard webcams without additional hardware.

Recent research demonstrates that vision-based gesture recognition systems can effectively replace traditional input devices and provide improved accessibility and interactive user experiences.

III. PROPOSED SYSTEM

The proposed Virtual Mouse system is an AI-powered software solution that allows users to control computer mouse operations using hand gestures captured through a webcam.

The system continuously detects hand landmarks and interprets specific gestures to perform corresponding mouse actions. MediaPipe Hand Tracking is used for landmark detection, while OpenCV processes video frames in real time.

Key Features of the Proposed System

Gesture-Based Cursor Control

The cursor movement is controlled by tracking the position of the user's index finger.

Touch-Free Interaction

The system enables users to operate computers without physically touching input devices.

AI-Based Hand Tracking

The integration of MediaPipe and OpenCV provides accurate and real-time gesture recognition.

Customizable Gesture Operations

Additional gestures can be programmed to perform various system operations.

User-Friendly GUI

A Tkinter-based interface is provided for easy system activation and control.

Hardware Independence

The system requires only a standard webcam and does not depend on external sensors or controllers.

Advantages of the Proposed System

- Provides a hygienic and contactless interaction method.
- Eliminates the need for expensive external hardware.
- Enhances accessibility for physically challenged users.
- Offers innovative interaction methods for gaming and virtual reality applications.
- Supports automation and smart environment control systems.

3.1 Block Diagram Description

The output section of the system represents the actions generated after processing hand gestures. The detected gestures are converted into mouse operations and displayed visually on the screen.

Cursor Movement Output

The cursor position changes according to the movement of the index finger.

Click Operation Output

Different pinch gestures are used to perform left-click and right-click operations.

Scrolling Output

Vertical movement of fingers enables scrolling functionality.

Drag-and-Drop Output

Continuous pinch gestures initiate drag-and-drop actions.

Brightness Control Output
The system adjusts screen brightness based on the distance between fingers.

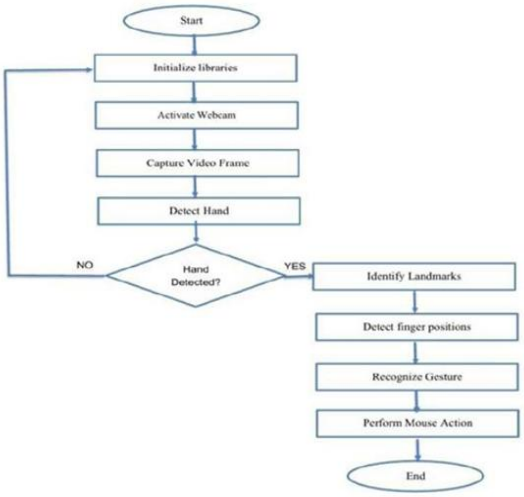
Visual Feedback Output
Detected hand landmarks and gesture recognition status are displayed on the webcam feed.

Real-Time System Response
The system processes gestures continuously with minimal delay to ensure smooth interaction.

3.2 Data Flow Description
Step 1: System Initialization
The application is launched through the Tkinter graphical user interface. The webcam is initialized, and the MediaPipe hand tracking model is loaded.
Step 2: Capturing Hand Gestures
OpenCV continuously captures video frames from the webcam. The frames are converted from BGR format to RGB format for MediaPipe compatibility.
Step 3: Hand Landmark Detection
MediaPipe detects 21 hand landmarks and extracts relevant finger positions required for gesture recognition.
Step 4: Gesture Processing
The detected landmark positions are processed using NumPy interpolation to map hand coordinates to screen coordinates.
Step 5: Mouse Action Execution
Based on gesture detection, PyAutoGUI performs the following actions:

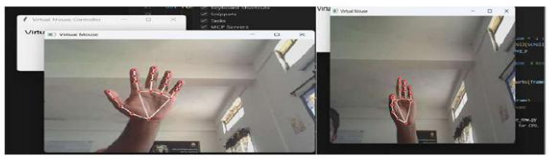
- Cursor Movement using index finger tracking
- Left Click using thumb-index pinch gesture
- Right Click using index-middle finger pinch gesture
- Scrolling using finger movement
- Drag-and-Drop using sustained pinch gestures

Step 6: Visual Feedback
The webcam feed displays detected hand landmarks and gesture status to provide user feedback.
Step 7: System Shutdown
The user can terminate the application using the GUI exit button or by pressing the ‘q’ key. The webcam resources are then safely released.

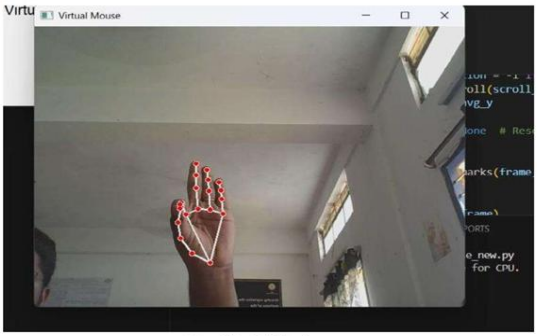


IV. OUTPUTS

The system successfully performs real-time cursor movement and gesture-based mouse operations using hand tracking technology. Experimental results demonstrate smooth cursor control, accurate gesture recognition, and efficient interaction with minimal latency.
The proposed system effectively replaces traditional mouse functionality while providing a modern touchless interaction experience.


Drag and Drop


Scroll



Left Click

V. CONCLUSION

The Virtual Mouse Using Hand Gesture Recognition and Computer Vision system successfully demonstrates an intelligent and touch-free alternative to conventional mouse devices. By integrating OpenCV, MediaPipe, and PyAutoGUI, the system enables accurate real-time hand tracking and efficient gesture-based cursor control.

The proposed system provides multiple mouse functionalities including cursor movement, clicking, scrolling, and drag-and-drop operations without requiring additional hardware devices. The system enhances accessibility for disabled users and improves hygiene in shared environments such as hospitals and public workspaces.

The lightweight and cost-effective design makes the system suitable for various applications including gaming, smart homes, automation, and virtual reality systems. Future enhancements may include multi-hand gesture support, advanced AI-based gesture classification, voice integration, and improved gesture customization.

REFERENCES

[1] "Gesture-Based Human Computer Interaction Using Computer Vision," IEEE Xplore.
[2] R. Szeliski, Computer Vision: Algorithms and Applications. Springer, 2011.
[3] W. McKinney, Python for Data Analysis. O'Reilly Media, 2017.
[4] "PlantUML Documentation." Available: PlantUML Official Website
[5] "NumPy Documentation." Available: NumPy Official Documentation
[6] "Python Documentation." Available: Python Official Documentation
[7] "MediaPipe Hand Tracking." Available: Google MediaPipe Documentation
[8] "OpenCV Library." Available: OpenCV Official Website
[9] "PyAutoGUI Documentation." Available: PyAutoGUI Documentation
[10] "Tkinter GUI Programming." Available: Tkinter Python Documentation
[11] 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, 2020.
[12] J. Qi, L. Ma, Z. Cui, et al., "Computer Vision-Based Hand Gesture Recognition for Human-Robot Interaction: A Review," Complex & Intelligent Systems, Springer, 2023.
[13] D. Sarma and M. K. Bhuyan, "Methods, Databases and Recent Advancement of Vision-Based Hand Gesture Recognition for HCI Systems," SN Computer Science, 2021.
[14] M. V. Gore and A. R. Karwankar, "Human Computer Interaction Using Hand Gesture Recognition," International Journal of Engineering Research & Technology (IJERT), 2014.
[15] J. Stephan and S. A. Jadwaa, "Gesture Recognition for Human-Computer Interaction," International Journal of Advancements in Computing Technology, 2010.