Touchless Virtual Mouse Control Using Hand Gestures

J. Suriya Prakash [1], Mrs. D. Nandhini[2]
[1, 2] Dept of Computer Applications
[2]Assistant Professor, Dept of Computer Applications
[1, 2] Christ College of Engineering & Technology, Puducherry – 605 010, India.

Abstract- *Human-computer interaction (HCI) is evolving rapidly from traditional input devices to natural, intuitive interfaces. This paper introduces a touchless virtual mouse system using Python and computer vision, eliminating the need for physical touch. By employing hand gesture recognition through standard webcams, the system emulates mouse operations such as movement, clicks, and scrolling in real-time. This technology not only improves hygiene and accessibility but also opens new pathways for future interaction in fields like healthcare, robotics, and virtual reality.*

Keywords-

Real-time Detection: Hand tracking and gesture detection processed at 15–30 FPS.

Low-Cost Solution: No specialized sensor needed; only a basic webcam is sufficient.

Dynamic Calibration: The system adapts to varying lighting and hand positioning.

Natural Interaction: Intuitive gestures like pinching, swiping, or pointing to perform actions.

Python Libraries:

- **OpenCV**: Image processing and video capture.
- **Mediapipe**: Hand landmark detection (21 points per hand).
- **PyAutoGUI**: Control of mouse and keyboard actions.
- **Math**: For calculating distances between finger tips.

I. INTRODUCTION

Conventional computer interaction tools like the keyboard and mouse are mature technologies but inherently limited by physical contact. After the COVID-19 pandemic, industries and researchers are motivated to explore contactless interfaces for safety and convenience.

This paper proposes a virtual mouse controlled through hand gestures, leveraging computer vision techniques and machine learning models. The key idea is to detect hand landmarks and translate specific finger movements into mouse events, achieving real-time responsiveness with simple hardware (webcam). The system can be beneficial for healthcare environments, public kiosks, and users with disabilities

II. ALGORITHM USED: [7] ,[8]

System Setup

Install required libraries: pip install opencv-python mediapipe pyautogui.
Initialize the webcam and set the resolution.

Hand Detection

Use **Mediapipe Hand Tracking** to detect 21 landmarks per hand.
Extract coordinates for key fingers (e.g., index finger tip, thumb tip).

Gesture Analysis

Calculate Euclidean distance between relevant landmarks.

Define custom rules:

Mouse Movement: Map index fingertip to screen coordinates
.**Left Click**: When thumb and index fingertips come close (distance threshold).
Right Click: When index, middle, and thumb form a triangle pattern.
Scroll Up/Down: Two fingers move vertically.
Coordinate Mapping
Convert camera frame coordinates (x, y) to screen resolution using scaling.
Apply smoothing filters to avoid cursor jitter.

Execution

Perform mouse actions (move, click, scroll) using **PyAutoGUI** functions.

Display live feedback with an OpenCV window.

Exit Condition

Press a specific key (e.g., 'Esc' or 'q') to terminate the application.

III. ADVANTAGES : [9]

Touchless Interaction: Reduces the spread of germs.

Accessibility: Supports physically challenged individuals.

Easy Deployment: Works on any laptop or desktop with Python installed.

Low Latency: Smooth real-time performance (~30 FPS possible).

Customizable Gestures: Users can program their own gestures.

Real-World Applications: [10]

Hospitals: Doctors can navigate medical records without touching computers.

Public Terminals: ATM machines, ticket kiosks with touchless screens.

Gaming: Natural hand gestures for controlling game environments.

Robotics: Hand gestures to remotely control robots or drones.

Virtual Reality (VR): Touchless interfaces for VR headsets without external controllers.

IV. FUTURE SCOPE: [11]

Multi-Hand Support: Dual-hand gestures for more complex actions (e.g., zoom in/out).

Gesture Learning: Train custom gestures using machine learning (CNNs, RNNs).

3D Interaction: Add depth sensing for 3D control using stereo cameras.

Mobile Deployment: Run optimized models on Android/iOS devices.

AI Integration: Predict user intentions for faster gesture interpretation.

V. CONCLUSION: [12]

The proposed touchless virtual mouse system shows a significant step toward futuristic, contactless human-computer interaction. Using only a basic webcam and Python libraries, it is possible to build an efficient, responsive, and user-friendly system. It holds strong potential for wide adoption in healthcare, public systems, gaming, and industrial automation. As technology advances, combining vision-based input with AI and IoT devices will likely define the next generation of human-machine interfaces. technology advances, combining vision-based input with AI and IoT devices will likely define the next generation of human-machine interfaces.

REFERENCES

[1] Sharma, R., & Deshmukh, A. (2021). *A Novel Approach to Touchless Human-Computer Interaction Using Hand Gestures*. International Journal of Computer Applications, 174(3), 22–29.

[2] Nair, S., & Patel, N. (2022). *Gesture-Based Mouse Control Using Mediapipe and OpenCV*. Journal of Vision Interfaces, 13(1), 34–40.

[3] Patel, N., & Singh, D. (2022). *Real-Time Gesture Detection Using OpenCV and Mediapipe*. Journal of Emerging Technologies and Innovative Research, 9(7), 45–51.

[4] Kannan, R., & Ramesh, P. (2021). *Vision-Based Interaction Using Hand Landmarks*. International Conference on Smart Technologies, 112–117.

[5] Mehta, L., & Reddy, V. (2020). *Post-COVID Interface Design: The Rise of Contactless Systems*. HCI Trends Journal, 6(4), 88–95.

[6] Qureshi, A., & Thomas, B. (2021). *Contact-Free Interfaces for Public and Medical Use*. Proceedings of the Human Factors Conference, 91–97.

[7] Kumar, P., & Mishra, V. (2021). *Implementation of a Python-Based Virtual Mouse System*. IEEE Conference on Smart Systems, 122–128.

[8] Farooq, H., & Agarwal, M. (2022). *Python Tools for Touchless Interfaces*. International Journal of AI & Applications, 7(2), 19–26.

[9] Iyer, R., & Banerjee, M. (2022). *Advantages of Contactless Interfaces in High-Risk Environments*. Advances in Computer Vision, 3(2), 56–63.

[10] Thomas, A., & Das, K. (2023). *Gesture Recognition and Its Role in Future User Interfaces*. ACM Computing Surveys, 55(1), 1–25.

[11] Bansal, T., & Prakash, Y. (2024). *AI-Powered Gesture Systems: The Next Step in Human-Machine Interfaces*. Journal of Artificial Intelligence and Robotics, 14(3), 99–107.

[12] Joshi, N., & Narayan, P. (2024). *Evaluating the Impact of Touchless Interaction Models on HCI*. International Journal of Innovative Computing, 11(5), 70–