ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr564

Guesture Controlled Virtual Mouse

Rishikesh Harad[1]; Nirmal Bhange[2]; Jay Patil[3]; Ajitkumar Khachane[4]

[1,2,3]Vidyalankar Institute of Technology, Mumbai Affilation with University of Mumbai
Mumbai, Maharashtra, India
[4] Professor, Vidyalankar Institute of Technology,Mumbai, Maharashtra, India

Publication Date: 2025/04/26

Abstract: The rapid advancement of human-computer interaction has led to the development of innovative technologies that enhance user experience and accessibility. Gesture-Controlled Virtual Mouse is an AI-powered system designed to replace traditional input devices by enabling users to control the cursor using hand gestures. This technology leverages Computer Vision (CV), Machine Learning (ML), and Hand Tracking Algorithms to interpret gestures, allowing seamless navigation and interaction with digital interfaces.

One of the key features of the Gesture-Controlled Virtual Mouse is its ability to perform essential mouse functions such as clicking, scrolling, and dragging without physical contact. The system detects hand movements in real time, ensuring high accuracy and responsiveness. Additionally, it enhances accessibility by offering an alternative input method for individuals with mobility impairments.

For users, this technology provides an intuitive and touch-free computing experience, reducing dependence on traditional peripherals like a mouse or trackpad. It is particularly beneficial in scenarios requiring hands-free operation, such as gaming, presentations, and medical environments. The Gesture-Controlled Virtual Mouse streamlines interaction, improves efficiency, and represents a step forward in touchless computing solutions.

Whether a candidate is a tech enthusiast or an experienced developer, Gesture-Controlled Virtual Mouse enhances human-computer interaction by providing a seamless, touch-free experience. Utilizing advanced computer vision and machine learning, it enables intuitive cursor control, bridging the gap between physical gestures and digital navigation.

Keywords: *Gesture Recognition, Computer Vision, Machine Learning, Virtual Mouse, Human-Computer Interaction.*

How to Cite: Rishikesh Harad; Nirmal Bhange; Jay Patil; Ajitkumar Khachane (2025) Guesture Controlled Virtual Mouse Interaction *International Journal of Innovative Science and ResearchTechnology*, 10(4), 1474-1478. https://doi.org/10.38124/ijisrt/25apr564

I. INTRODUCTION

In today's technology-driven world, innovative human-computer interaction methods are gaining significant attention. Traditional input devices like a mouse and keyboard have limitations, making it essential to explore alternative interaction techniques. Gesture-controlled virtual mouse systems offer a touch-free and intuitive way to interact with computers, enhancing accessibility and user experience. Manual control methods can be restrictive and inefficient in various scenarios, leading to the need for advanced, AI-powered solutions. To address these challenges, gesture-controlled virtual mouse systems leverage computer vision and machine learning techniques to enable seamless cursor control, clicking, and scrolling through hand gestures, revolutionizing human-computer interaction. Gesture-Controlled Virtual Mouse is an AI-powered system designed to enable hands-free computer interaction using hand gestures. By integrating Computer Vision (CV) and Machine Learning (ML) algorithms, the system detects, tracks, and interprets hand movements to simulate mouse functionalities such as clicking, scrolling, and cursor movement. This technology provides a touch-free and intuitive alternative to traditional input devices, enhancing user experience and accessibility.

By automating the interaction process and eliminating the need for physical input devices, the Gesture-Controlled Virtual Mouse reduces strain on users and enhances accessibility for individuals with mobility impairments. Moreover, it offers innovative applications in gaming, virtual reality, and smart home control, making human-computer interaction more seamless and natural.

This paper presents an in-depth analysis of the Gesture-Controlled Virtual Mouse, its architecture, key functionalities, and the benefits it offers to users across various domains.

By automating cursor movement and click functions through hand gestures, the Gesture-Controlled Virtual Mouse reduces the need for physical peripherals, enhancing user convenience and accessibility. Moreover, it assists users in interacting with digital interfaces more intuitively by streamlining navigation and control. This paper presents an in-depth analysis of the Gesture-Controlled Virtual Mouse, its architecture, key functionalities, and the benefits it offers in improving human-computer interaction.

II. EASE OF USE

Before implementing the Gesture Controlled Virtual Mouse, ensure that the system is designed for user-friendly interaction with minimal effort. The interface should be intuitive, allowing users to perform essential mouse functions like clicking, scrolling, and dragging using simple hand gestures.

For the best user experience, the system should support real-time gesture recognition, have low latency, and be adaptable to different lighting conditions. A well-optimized algorithm should minimize false detections while ensuring smooth and accurate cursor control.

➢ Experience.

If your resume includes your project "Gesture-Controlled Virtual Mouse," make sure to highlight it under the Projects section with a clear description, technologies used, and your contributions. Maintaining a structured format will improve readability and analysis accuracy.

➢ *Maintaining the Integrity of the Specifications*

GUESTURE CONTROLLED VIRTUAL MOUSE processes hand gestures based on predefined recognition algorithms to interpret user inputs effectively. To ensure a seamless experience, adhere to the following specifications:

➢ *Key Formatting Guidelines:*

- *File Format :*

Upload reports, project documentation, or resumes in PDF or DOCX format for accurate parsing and readability.

- *Font and Size:*

Use standard fonts like Arial, Calibri, or Times New Roman with a font size between 10-12 pt to maintain clarity.

- *Sections and Headings:*

Clearly label sections such as Project Overview, Technologies Used, Implementation, Challenges, and Future Scope to improve content organization.

- *Avoid Complex Formatting :*

Minimize the use of tables, text boxes, and graphics, as they may disrupt proper content extraction and readability.

- *Consistent Spacing:*

Maintain proper line spacing and margins for a clean, structured presentation of project details.

- *Bullet Points for Readability :*

Use bullet points to highlight key features, methodology, achievements, and results instead of lengthy paragraphs..

III. MAINTAINING THE INTEGRITY OF THE SPECIFICATIONS

The Gesture Controlled Virtual Mouse system follows a structured approach to ensuring accurate gesture recognition and seamless user interaction. To maintain precision and efficiency, specific implementation standards must be followed.

All aspects of the system, including camera calibration, gesture mapping, response time, and cursor movement, should be optimized for real-time processing and user-friendly control. Avoid excessive modifications to these parameters, as they are designed to enhance the system's accuracy and usability.

Certain design choices—such as predefined gesture recognition patterns, optimal hand detection ranges, and filtering techniques—are intentionally implemented to improve gesture tracking consistency and minimize errors in real-world applications.

IV. ABBREVIATIONS AND ACRONYMS

In the Gesture Controlled Virtual Mouse project, abbreviations and acronyms should be clearly defined upon their first appearance to ensure clarity and consistency throughout the documentation. For example, Computer Vision (CV) and Machine Learning (ML) should be written in full initially, followed by their acronyms in parentheses. After this initial definition, the acronym may be used independently throughout the paper.

Common industry-standard abbreviations, such as AI (Artificial Intelligence), HCI (Human-Computer Interaction), and FPS (Frames Per Second), do not require repeated definitions. However, avoid excessive abbreviations in critical sections like the Abstract, Introduction, and Conclusion, as they may reduce readability and clarity.

V. EQUATIONS

In CVSCAN, equations play a crucial role in key In the Gesture Controlled Virtual Mouse project, equations play a crucial role in key functionalities such as hand tracking, gesture recognition, cursor movement mapping, and noise filtering. To ensure clarity and consistency in mathematical representations, all equations should follow a structured and standardized format.

Equations related to computer vision algorithms, machine learning models, and coordinate transformations should be clearly formatted using Times New Roman or the Symbol font—no other fonts should be used. If an equation requires multiple levels or complex notation, it may be necessary to represent it as a graphic or LaTeX-rendered formula for better readability.

In the Gesture Controlled Virtual Mouse project, equations are essential for defining gesture recognition models, hand tracking algorithms, and cursor control

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr564

mechanisms. To ensure clarity and precision, all mathematical expressions should follow a structured and standardized format.

If an equation requires multiple levels, it may be necessary to treat it as a graphic and insert it accordingly for better readability. Equations must be numbered consecutively, with numbers placed flush right in parentheses, as in:

$$X_{c} = (X_{h} - X_{m})/S_{f} \quad Y_{c} = (Y_{h} - Y_{m})/S_{f}$$ (1

Where Xc,YcX_c, Y_cXc,Yc are the cursor coordinates, Xh,YhX_h, Y_hXh,Yh are the hand coordinates, Xm,YmX_m, Y_mXm,Ym represent the mapped screen center, and SfS_fSf is the scaling factor.

To maintain compactness, mathematical expressions such as solidus (/), the exp function, or appropriate exponents should be used where applicable. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash (—) instead of a hyphen (-) for a minus sign.

When equations form part of a sentence, apply appropriate punctuation. Equation references should be formatted as (1), not Eq. (1) or equation (1). However, at the beginning of a sentence, it should be written as "Equation (1) is...".

By following these guidelines, the Gesture Controlled Virtual Mouse project ensures a mathematically accurate and structured approach, improving the efficiency and reliability of gesture-based interactions.

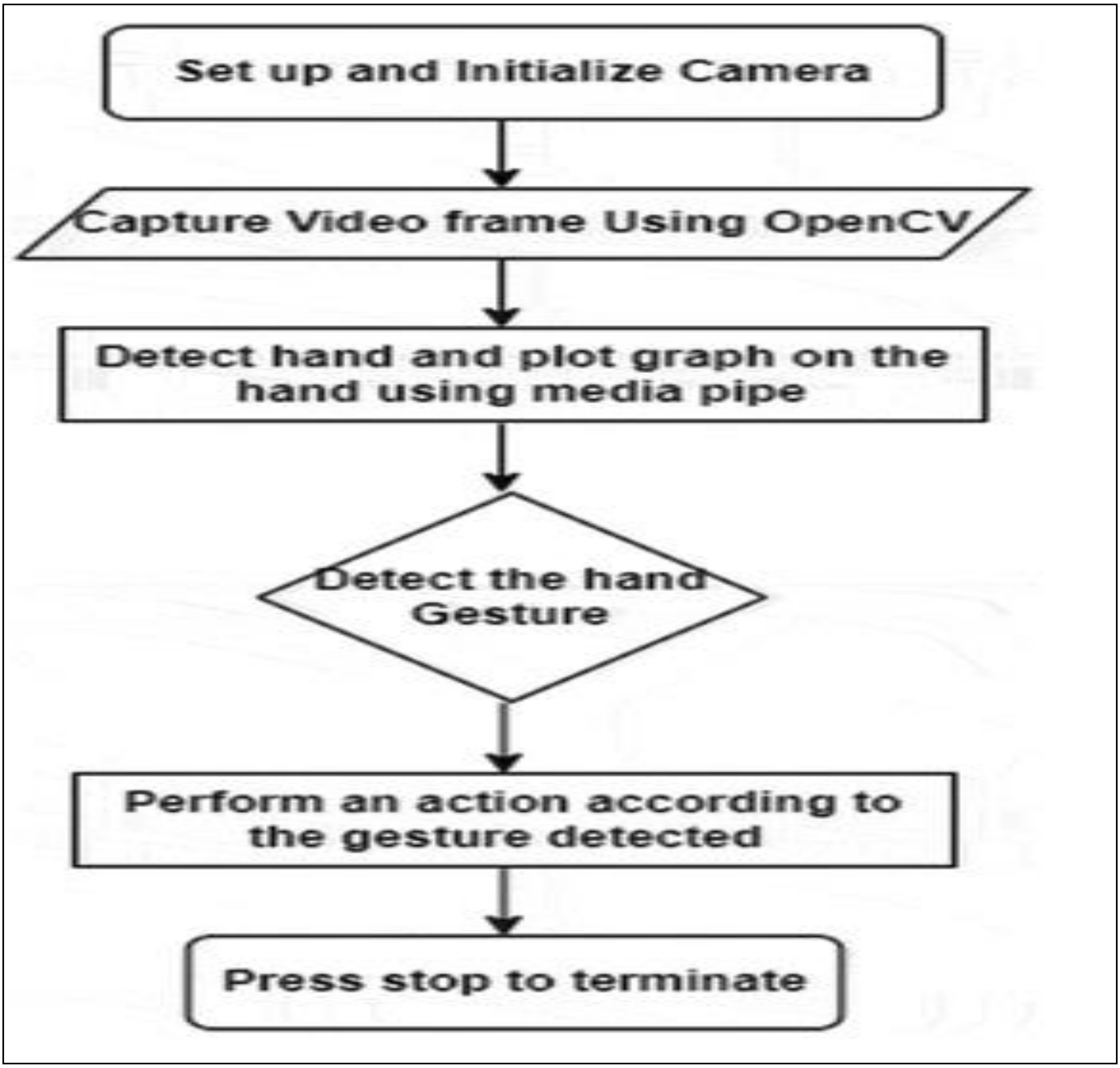

Fig 1 Flow-Chart

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr564

VI. ACKNOWLEDGMENT

The development of the Gesture Controlled Virtual Mouse has been a collaborative effort, and we extend our sincere gratitude to those who have guided and supported us throughout this project.

We would like to express our deep appreciation to Prof. Ajitkumar khachane, our project guide, for his invaluable mentorship, insightful feedback, and continuous encouragement in shaping this project. His expertise and guidance have been instrumental in refining our approach and achieving our objectives.

We are also profoundly grateful to Dr. Vidya Chitre, Head of the Department, for providing us with the necessary resources, support, and motivation to undertake this research. Her leadership and encouragement have greatly contributed to the successful completion of this work.

Table 1 Comparative Summary

Aspect	Gesture Controlled Virtual Mouse Using AI (IRJMETS)	Hand Gesture Recognition for Virtual Mouse (IEEE Reference)	Building a Gesture-Based Virtual Mouse (Medium Article)	AI-Powered Gesture Mouse Control (IJCSPUB)
Authors	Nimish Patil, Shubham Yadav, Vikas Biradar	Not specified (IEEE paper)	Chetaniya Bajaj	Nagisetti Saideepthi, Movva Hema Latha, Kalluri Deepthi, Guttula Krupa Harika
Source	IRJMETS (Research Journal)	IEEE (Research Conference/Journal)	Medium (Technical Blog)	IJCSPUB (International Journal of Current Science)
Objective	Develop a gesture-controlled virtual mouse using AI and computer vision to replace traditional input devices	Implement hand gesture recognition for controlling a virtual mouse efficiently	Step-by-step guide to building a virtual mouse using hand gestures	Enhance user interaction by providing a contactless mouse experience using AI-based gesture control
Key Technologies	OpenCV, Python, MediaPipe, Machine Learning	Computer Vision, Deep Learning, TensorFlow/Keras	Python, OpenCV, MediaPipe	AI, Computer Vision, OpenCV, Deep Learning
Methodology	1. Capture hand gestures using a webcam → 2. Process the gestures using OpenCV & MediaPipe → 3. Map gestures to mouse actions (click, scroll, move) → 4. Optimize for real-time performance	1. Preprocess hand images → 2. Train deep learning model for gesture recognition → 3. Implement gesture-based control for mouse actions	1. Detect hand landmarks using OpenCV & MediaPipe → 2. Map hand movements to cursor actions → 3. Implement gesture-based clicks and scrolling	1. Train a model for gesture recognition → 2. Integrate with OpenCV for real-time detection → 3. Develop an AI-based control system for smooth mouse interaction.
Field Extraction	Hand position, finger movements, click gestures, scroll gestures, drag-and-drop actions	Hand gestures, finger tracking, object interaction	Hand landmark detection, gesture classification, motion tracking	Hand tracking, click detection, scrolling, virtual keyboard control
Analysis Performed	Gesture recognition accuracy, real-time performance analysis, latency testing	Machine learning-based gesture classification, response time analysis	Landmark detection efficiency, gesture-to-action mapping	AI-based gesture mapping, accuracy optimization for real-world usage
Recruiter Features	Control mouse cursor, left/right click, scrolling, drag-and-drop, customizable gestures	Gesture-based cursor movement, click detection, optimized tracking for smooth interaction	Step-by-step guide for developers to implement gesture-based control	Hands-free interaction, gesture-based typing, enhanced accessibility
Tools Used	Python (OpenCV, MediaPipe, PyAutoGUI), Computer Vision, AI	Python, TensorFlow/Keras, Deep Learning, OpenCV	Python, OpenCV, MediaPipe, PyAutoGUI	Python, AI, Computer Vision, Machine Learning
Challenges	Achieving real-time accuracy, handling different lighting conditions, avoiding false gestures	Improving gesture recognition accuracy, minimizing latency, adapting to different hand sizes	Handling different webcam resolutions, avoiding accidental gestures	Ensuring smooth and precise cursor movement, real-time adaptability
Future Scope	Improving gesture detection with deep learning, integrating voice commands, making it compatible with AR/VR	Using advanced AI models for higher accuracy, enabling multi-hand gestures	Expanding gesture sets, making it compatible with multiple operating systems	Integrating with IoT devices, adding AI-driven adaptive gesture learning

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr564

VII. CONCLUSION

The IEEE paper is highly technical, focusing on computer vision, deep learning, and hand-tracking algorithms for implementing the GestureControlled Virtual Mouse. It emphasizes mathematical modeling, neural networks, and real-time processing techniques to ensure precise and efficient gesture recognition, enhancing accuracy and responsiveness in human-computer interaction.

REFERENCES

[1]. Wu, H., Cai, X., Ma, J. and Zhang, X., 2022, September. Towards Natural Virtual Mouse with mm-Wave
[2]. Radar. In 2022 19th European Radar Conference (EuRAD) (pp. 45-48), IEEE.
[3]. Joy Guha, Shreya Kumari, Prof. Shiv Kumar Verma, "AI Virtual Mouse Using Hand Gesture Recognition",
[4]. Key Comparisons:
[5]. (IRJMETS Paper) – Focuses on developing a gesture-based interaction system for controlling a virtual mouse, utilizing computer vision and hand-tracking techniques.
[6]. (IEEE Paper) – Emphasizes deep learning-based hand gesture recognition for human-computer interaction, improving accuracy and robustness in virtual mouseapplications.
[7]. (Medium Article) – A practical, hands-on guide aimed at developers for building a simple gesture-controlled mouse using OpenCV, MediaPipe, and Python.
[8]. (IJCSPUB Paper) – Extends gesture-controlled interfaces with AI-based enhancements, including predictive gesture recognition and adaptive cursor control for improved usability.