International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 10, Issue 01, ISSN No. 2455-2143, Pages 76-81
Published Online May 2025 in IJEAST (http://www.ijeast.com)

GESTURE CONTROL MOUSE USING MACHINE LEARNING

Shriram Surse, Pankaj Koli, Tahoor Khan
MIT School of Computing
MIT ADT University
Loni Kalbhor, Pune.

Prof. Aditi Wangikar
Computer science Engineering MITSOC
MIT ADT University, Loni kalbhor,
Pune, India.

Abstract—**In this research paper, we present a pioneering system for gesture-based control of a virtual mouse using computer vision and machine learning techniques. The system leverages the capabilities of the MediaPipe library and OpenCV to accurately detect and track hand gestures in real time through a standard webcam. The virtual mouse controller, implemented in Python, interprets the hand gestures to enable intuitive and natural interaction with the computer.**

Our approach includes the detection and tracking of multiple hand landmarks, allowing for precise mapping of hand movements and finger positions. The system recognizes various gestures, such as right swipe, left swipe, screenshots, left and right clicks, double-clicks, scrolling, zooming, and dragging, enhancing the user experience in virtual environments.

The gestures are translated into corresponding mouse actions using the PyAutoGUI library, enabling seamless integration with existing applications. The gesture control system is characterized by its adaptability to diverse hand gestures and robust performance in different lighting conditions. The controller's functionality encompasses right swipe, left swipe, screenshots, cursor movement, scrolling, zooming in and out, left and right clicking, double-clicking, and dragging.

Additionally, the system introduces a dynamic and responsive cursor movement algorithm, enhancing user control and precision. Our research contributes to the evolving field of human-computer interaction by providing an innovative and practical solution for hands-free computer control.

The gesture-controlled virtual mouse offers potential applications in various domains, including accessibility technology, virtual reality, and interactive presentations. The paper concludes with a discussion of potential future enhancements and extensions, emphasizing the system's versatility and its impact on the evolution of intuitive human-computer interfaces.

Keywords—**Computer Vision, Machine Learning, Mediapipe, OpenCV, PyAutoGUI, Hand Tracking, Flask**

I. INTRODUCTION

The landscape of human-computer interaction (HCI) has undergone a transformative shift towards more intuitive interfaces, with gesture-based control systems standing out as a promising avenue. Our research introduces a pioneering approach to HCI through the development of a virtual mouse controlled by gestures. At the heart of our system is the integration of computer vision and machine learning technologies, specifically making use of the MediaPipe library and OpenCV.

These tools empower real-time detection and tracking of hand gestures through a standard webcam, providing users with a hands-free and immersive computing experience.. Implemented in Python, the virtual mouse controller interprets intricate hand movements, translating them into a variety of computer actions. Going beyond standard hand tracking, our system recognizes multiple hand landmarks, facilitating precise mapping of finger positions.

This level of detail allows for the identification of diverse gestures, including right swipe, left swipe, screenshots, left and right clicks, double-clicks, scrolling, zooming, and dragging. A notable feature of our approach is its adaptability to various hand gestures and its robust performance under different lighting conditions.

The system's capabilities include right swipe, left swipe, screenshots, cursor movement, scrolling, zooming in and out, left and right clicking, double clicking, and dragging. Additionally, a dynamic and responsive cursor movement algorithm enhances user control and precision.

This research paper aims to contribute to the evolving field of HCI by presenting an innovative and practical solution for hands-free computer control. The gesture-controlled virtual mouse not only opens new possibilities for accessibility technology but also finds applications in virtual reality interactions and interactive presentations. The paper concludes with a discussion on potential future enhancements, emphasizing the system's versatility and its impact on advancing intuitive human computer interfaces.

II. LITERATURE REVIEW

[1] This research focuses on creating virtual mouse using computer vision which delivers good results in hand gestures but yet has some less features and not so accurate results.

[2] This review focuses on the development of a hardware-based system. While the model delivers highly accurate results, executing certain movements becomes challenging when wearing a glove, significantly limiting the user's hand range of motion, speed, and agility. Prolonged glove usage may also lead to skin diseases, making it less suitable for users with sensitive skin types.

[3] The authors introduced a machine-user interface employing computer vision and multimedia techniques for hand gesture detection. However, a notable drawback is the requirement for skin pixel identification and hand segmentation from stored frames before applying gesture comparison techniques.

[4] This study presents a system for recognizing hand movements using a mobile phone's camera and a connected mobile projector for visual feedback. The suggested architecture allows easy integration of mobile applications with their framework for gesture recognition. This approach facilitates the rapid creation of research prototypes, diverting the user's focus from the device to the content.

[5] A novel method is proposed for performing mouse functions without additional electrical equipment, relying solely on a webcam. Mouse actions, including clicking and dragging files, are executed through hand gestures. However, the model exhibits lower accuracy and lacks extensive mouse functionality.

[6] The 2020 research on Virtual Mouse Using Object Tracking by Monali Shetty, Christina Daniel, Manthan Bhatkar, and Ofrin Lopez relies on computer vision and HSV colour recognition to capture hand gestures from a webcam. This technology enables users to control the system pointer and perform mouse actions using coloured caps or tapes tracked by the computer's webcam.

[7] Vantukal Reddy, Thumma Dhyanchand, Galla Vamsi Krishna, and Satish Maheshwaram's Virtual Mouse Control Using Colored Fingertips And Hand Gesture Recognition, published in 2020, utilizes fingertip detection and hand motion tracking through image-based live video. Two techniques, hand gesture detection and coloured caps, are employed to track fingers, enabling precise control of the virtual mouse through hand gestures.

[8] Real-Time Virtual Mouse System Using RGB-D Pictures And Fingertip Detection by Dinh SonTran, HyungJeong Yang, Ngoc-Huynh Ho Soo HyungKim, and Guee Sang Lee leverages Microsoft Kinect Sensor version 2 for hand recognition and segmentation. Utilizing fingertip sensing and RGB-D pictures, their system extracts detailed skeleton-joint information for real-time virtual mouse control, providing a revolutionary approach to hand interaction extended to include web interaction capabilities using Flask. This allowed users to interact with web applications and perform mouse actions through hand gestures.

III. METHODOLOGY

The proposed system uses web camera for capturing images or video based on the frames. For capturing we are using CV library Open cv which is belongs to python web camera will start capturing the video and Open cv will create a object of video capture. To AI based virtual system the frames are passed from the captured web camera.

The capturing of the frame was done with the AI virtual mouse system until the program termination. Then the video captured has to be processed to find the hands in the frame in each set. The processing takes places is it converts the BRG images into RGB images, which can be performed with the below code, image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR_BGR2RGB) image. flags. writeable = False results = hands. process(image) This code is used to flip the image in the horizontal direction then the resultant image is converted from the BRG scale to RGB scaled image.

4.3. Rectangular Region for Moving through the Window

The windows display is marked with the rectangular region for capturing the hand gesture to perform mouse action based on the gesture. when the hands are find under those rectangular area the detection begins to detect the action based on that the mouse cursor functions will be performed. The rectangular region is drawn for the purpose of capturing the hand gestures through the web camera which are used for mouse cursor operations.

International Journal of Engineering Applied Sciences and Technology, 2025
Vol. 10, Issue 01, ISSN No. 2455-2143, Pages 76-81
Published Online May 2025 in IJEAST (http://www.ijeast.com)

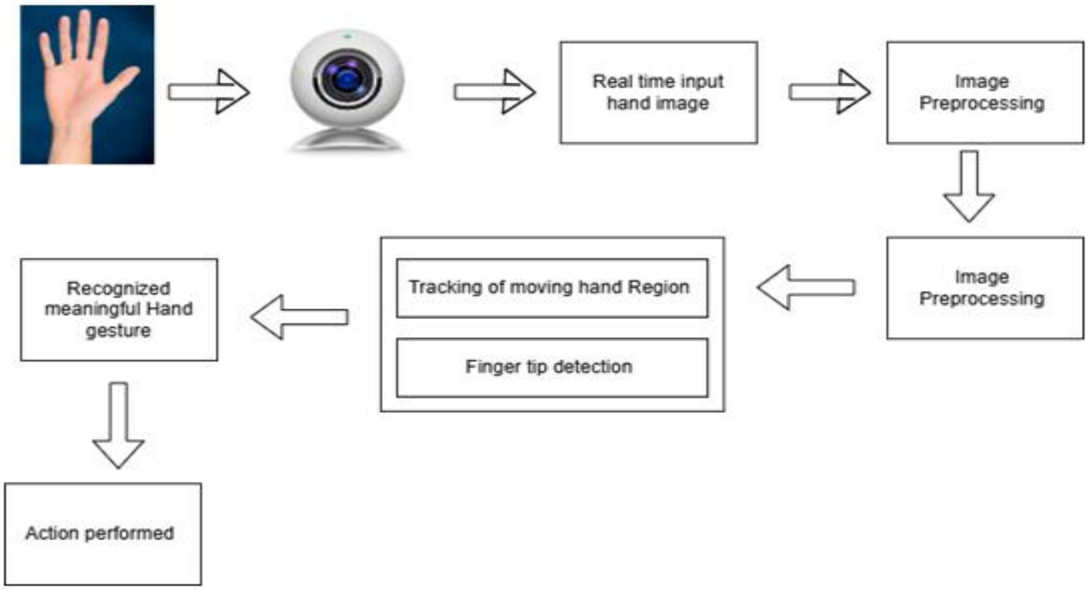

Fig 1. [Block diagram]

IV. SYSTEM DESIGN AND FEATURES

The following flowchart serves as an exceptionally effective tool for comprehending the system's architecture because it prioritizes conceptual clarity over technical complexity.

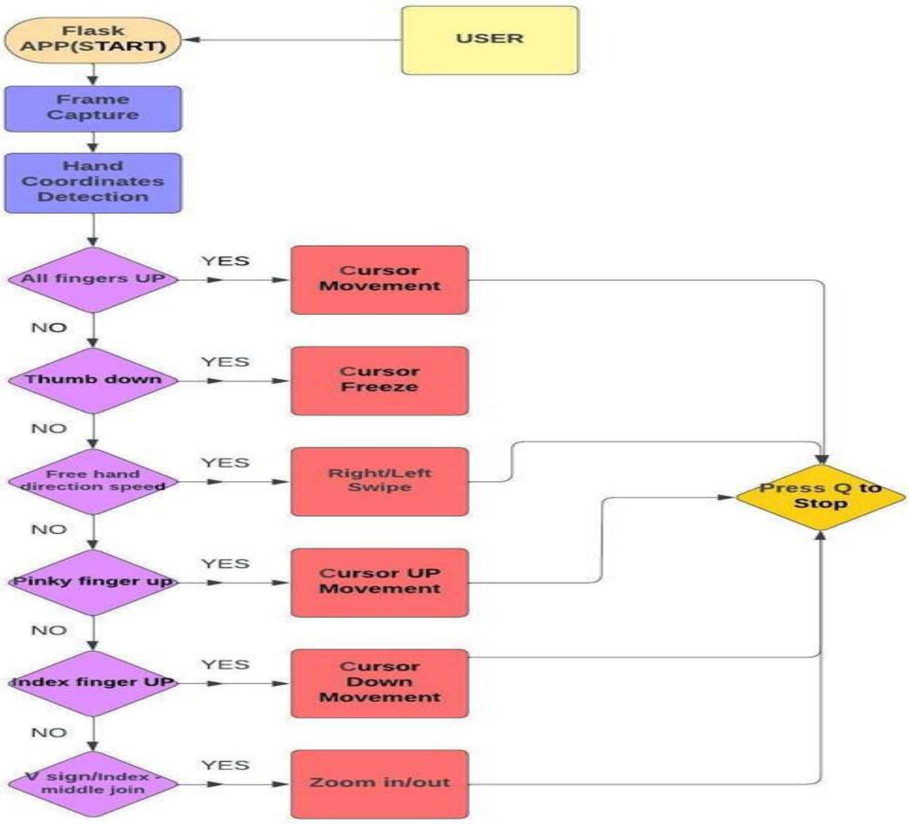

Fig. 1. Flowchart for virtual mouse using hands gestures.

1.Hand Gesture Recognition: - The system employs the MediaPipe library and OpenCV for accurate hand gesture detection and tracking. - Multiple hand landmarks are identified, providing detailed information about finger positions and hand movements. The proposed system is a gesture-controlled virtual mouse that redefines human-computer interaction by introducing a seamless and intuitive method for users to interact with digital environments.

The proposed system is a gesture-controlled virtual mouse that redefines human-computer interaction by introducing a seamless and intuitive method for users to interact with digital environments. Leveraging computer vision and machine learning technologies, the system utilizes the MediaPipe library and OpenCV to enable real-time detection and tracking of hand gestures through a standard webcam.

2. Virtual Mouse Controller: - Implemented in Python, the virtual mouse controller interprets recognized hand gestures and translates them into corresponding computer actions. - The controller recognizes a variety of gestures, including left and right clicks, double-clicks, scrolling, zooming, and dragging. 3. Adaptive Hand Tracking: - The system demonstrates adaptability to diverse hand gestures, ensuring robust performance. - Precise mapping of finger positions enhances the accuracy and responsiveness of the virtual mouse.

1. Cursor Movement: - The system allows users to control the cursor's position on the screen using hand movements.
2. Clicking and Double-Clicking: - Recognizes gestures for left and right clicks, as well as double-clicks, enabling seamless navigation and interaction with applications.
3. Scrolling: - Implements scrolling gestures for both upward and downward movements, enhancing user experience in applications that require vertical navigation.
4. Zooming: - Recognizes gestures for zooming in and out, providing users with an intuitive method for adjusting the scale of content.
5. Dragging: - Enables dragging functionality, allowing users to interact with elements on the screen by holding and moving them.
6. Screenshot: - Enables screenshot functionality, allowing users to take a screenshot and save it into the system.
7. Left/Right Swipe: - Helps the users to right or left swipe on the screen.cv2 (OpenCV) Purpose: OpenCV is a powerful computer vision library widely used for various image and video processing tasks.

cv2.VideoCapture(0): Initializes a video capture object for accessing the webcam.

cv2.flip(img, 1): Flips the video frame horizontally, commonly used for webcam input.

cv2.cvtColor(img, cv2.COLOR_BGR2RGB): Converts the color space of the image from BGR to RGB, a common step when working with certain computer vision libraries.

.

V. RESULT AND ANALYSIS

Recent experiments on gesture-controlled virtual mouse systems show promising results in gesture recognition and system performance, but challenges remain. One study using Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) achieved over 90% accuracy in recognizing basic gestures, such as pointing and clicking, in controlled environments. However, performance was hindered on low-resource devices, with response times lagging and computational demands increasing significantly during real-time processing.

The results of these experiments highlight the effectiveness of deep learning models in recognizing gestures with high accuracy, but they also underscore the need for further improvements in real-time processing speeds and environmental adaptability. Some studies found that the systems performed best under controlled conditions, but issues arose when the system was subjected to changing lighting or cluttered backgrounds [8].

Overall, while the systems demonstrate promising results, particularly in terms of gesture recognition accuracy, there is still room for improvement in reducing computational load, enhancing real-time processing, and increasing robustness against environmental fact

VI. FUTURE WORK

- User Experience: Gesture-controlled mice provide a hands-free, intuitive interaction method for users, enhancing accessibility and usability, particularly for individuals with disabilities.
- Industry Applications: Potential uses span healthcare (sterile environments), entertainment (gaming, VR/AR), and smart homes (IoT devices), improving interaction without physical devices.
- Assistive Technology: Offers an alternative input method for people with physical limitations, customizable to individual needs.
- Machine Learning Advancements: Real-time gesture recognition, improved personalization, and adaptive learning models can optimize performance, creating more seamless interactions.

Feasibility:

- Machine Learning & Computer Vision: Deep learning models, especially CNNs, and real-time processing (with edge computing) are crucial for accurate gesture recognition.
- Hardware: Cameras, depth sensors, and wearables (e.g., smart gloves) support gesture tracking, improving precision and reliability.
- Challenges: Issues include accuracy, latency, and ergonomics. Continuous training data and robust model optimization are necessary to improve reliability and reduce user fatigue.

OUTPUT

For the Mouse Cursor moving around the Computer Window.

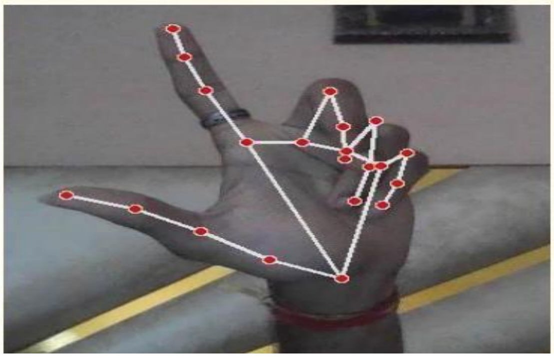

Figure 1: Computer Window with Mouse Cursor

- **To perform Drag & Drop Operation**

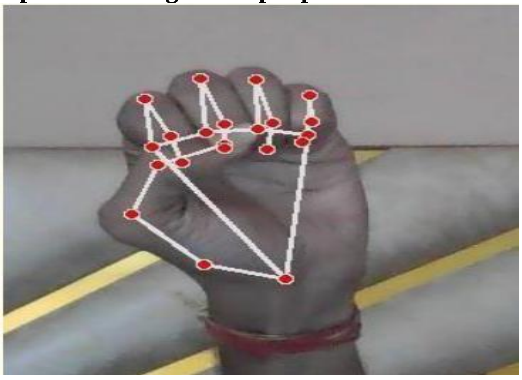

Figure 2: Mouse Operation-Drag & Drop

- **To Perform Left Button Click operation**

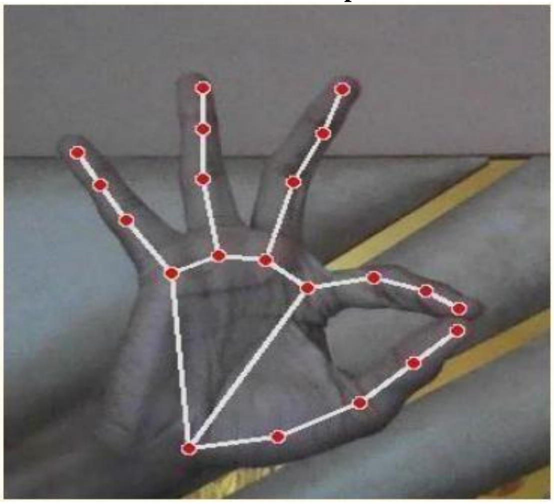

Figure 3: Mouse Operation- Left Click

VII. CONCLUSION

The incorporation of virtual mouse technology through hand gestures has garnered significant interest, offering a novel approach to device control that enhances accessibility, convenience, and hygiene. Despite its advantages, it's crucial to acknowledge limitations such as potential inaccuracies and increased physical effort required from users. Additionally, this technology may not be universally applicable across all applications or user groups. Nevertheless, the virtual mouse utilizing hand gesture technology shows immense promise as a valuable tool for various users, especially those facing challenges with conventional input devices. With ongoing advancements, this technology has the potential to revolutionize user interactions and cater to diverse needs in the digital landscape.

VIII. VIII. ACKNOWLEDGMENT

We extend our sincere gratitude to the faculty of MIT ADT University for their invaluable guidance, continuous support, and constructive feedback throughout the development of this project. Their expertise and encouragement have been instrumental in shaping this Gestured controlled virtual mouse. We are particularly thankful for their insights on technical implementation, ethical considerations, and user-centric design, which greatly enhanced the quality of our work. Special thanks to our mentors for their patience and dedication in helping us overcome challenges and refine our approach. This project would not have reached its current stage without their unwavering commitment to fostering innovation and academic excellence.

IX. REFERENCES

[1]. Quam, D.L., et.al. (1990). Gesture Recognition with a Dataglove. In IEEE conference on Aerospace and Electronics (pp. 755-760).

[2]. Guoli Wang., et.al. (2015). Optical Mouse Sensor-Based Laser Spot Tracking for HCI input, Proceedings of the Chinese Intelligent Systems Conference (pp. 329-340).

[3]. Baldauf, M., and Frohlich, p. (2013). Supporting Hand Gesture Manipulation of Projected Content with mobile phones. In the European conference on computer vision (pp. 381-390).

[4]. Roshnee Matlani., Roshan Dadlani., Sharv Dumbre., Shruti Mishra., & Abha Tewari. (2021). Virtual Mouse Hand Gestures. In the International Conference on Technology Advancements and innovations (pp. 340-345).

[5]. Mayur, Yeshi., Pradeep, Kale., Bhushan, Yeshi., & Vinod Sonawane. (2016). Hand Gesture Recognition for Human-Computer Interaction. In the international journal of scientific development and research (pp. 9-13).

[6]. Shriram, S., Nagaraj, B., Jaya, J., Sankar, S., & Ajay, P. (2021). Deep Learning Based Real-Time

AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread. In the Journal of Healthcare Engineering (pp. 3076-3083).

[7]. Steven Raj, N., Veeresh Gobbur, S., Praveen., Rahul Patil., & Veerendra Naik. (2020). Implementing Hand Gesture Mouse Using OpenCV. In the International Research Journal of Engineering and Technology (pp. 4257-4261).

[8]. Sneha, U., Monika, B., & Ashwini, M. (2013). Cursor Control System Using Hand Gesture Recognition. In the International Journal of Advanced Research in Computer and Communication Engineering (pp. 2278-1021).

[9]. Krishnamoorthi, M., Gowtham, S., Sanjeevi, K., & Revanth Vishnu, R. (2022). Virtual mouse using YOLO. In the international conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (pp. 1-7).

[10]. Varun, K.S., Puneeth, I., & Jacob, T.p. (2019). Virtual Mouse Implementation using OpenCV. In the International Conference on Trends in Electronics and Informatics (pp. 435-438).

[11]. Quek, F., et.al. (1994). Towards a vision based hand gesture interface, in Proceedings of Virtual Reality Software and Technology (pp. 17-31).

[12]. Tharsanee, R.M., Soundariya, R.s., Kumar, A.S., Karthiga, M., & Sountharrajan, S. (2021). Deep Convolutional neural network-based image classification for COVID-19 diagnosis. In Data Science for COVID-19 (pp. 117-145). Academic Press.

[13]. Newell, A., Yang, K., & Deng, J. (2016, October). Stacked hourglass networks for human pose estimation. In the European conference on computer vision (pp. 483-499). Springer, Cham.

[14]. Ramakrishna, V., Munoz, D., Hebert, M., Andrew Bagnell, J., & Sheikh, Y. (2014).Pose machines: Articulated pose estimation via inference machines. In the European Conference on Computer Vision (pp. 33-47). Springer, Cham.

[15]. Tharani, G., Gopikasri, R., Hemapriya R., & Karthiga, M. (2022). Gym Posture Recognition and Feedback Generation Using Mediapipe and OpenCV. In International Journal of Advance Research and Innovative Ideas in Education (pp. 2053-2057).

[16]. Shibly, K.H., Kumar, S., Islam, M.A., & Iftekhar Showrav, S. (2019). Design and Development of Hand Gesture Based Virtual Mouse. In the International Conference on Advances in Science, Engineering and Robotics Technology (pp. 1-5).