Gesture-Controlled Virtual Mouse with Computer Vision

[1]Om Lade, [2]Esha Elnoorkar, [3]Snehal Tornalkar,
[4]Anjali Artham, [5]Dr. Dhananjay M

[1234]Under Graduate, Department of AI&ML-GNDEC-Bidar
[5]Principal,GNDEC-Bidar

ABSTRACT:

The evolution of Human-Computer Interaction (HCI) has significantly transformed the way users interact with digital systems. Among these innovations, the computer mouse has played a crucial role in enhancing user experiences by providing seamless control over graphical user interfaces. However, traditional computer mice, whether wired or wireless, still rely on physical components such as batteries, dongles, and Bluetooth connectivity. These dependencies create limitations in terms of portability, accessibility, and flexibility. Addressing these challenges, we propose an AI-powered gesture-driven virtual mouse interface that leverages computer vision to eliminate the need for physical devices entirely.This research introduces a vision-based virtual mouse system that utilizes a webcam or built-in camera to detect hand gestures and fingertip movements, enabling users to interact with their computers without any additional hardware. By harnessing the power of MediaPipe, a real-time hand-tracking framework, our system accurately recognizes hand gestures and translates them into corresponding mouse actions. To further enhance functionality, we integrate PyInput, AutoPy, and PyAutoGUI libraries to facilitate cursor control and implement core mouse functionalities such as left-click, right-click, scrolling, and drag-and-drop operations. The result is a seamless, intuitive, and touch-free HCI system that mimics the functionality of a physical mouse while offering enhanced accessibility and convenience.

INTRODUCTION

With the rapid advancements in augmented reality and everyday technology, devices are becoming more compact and seamlessly integrated into our lives through Bluetooth and wireless technologies. Traditional computer peripherals, such as the mouse, require external hardware like a dongle for connectivity and batteries for power. However, what if we could eliminate these physical constraints and interact with our computers in a more intuitive way? This paper introduces an AI-powered virtual mouse system that leverages hand gestures and fingertip detection to perform essential mouse functions. Instead of relying on a traditional mouse, this system utilizes a computer's built-in camera or an external webcam to track hand movements and gestures, allowing users to control the cursor, click, scroll, and navigate their computers effortlessly.At the core of this system lies computer vision technology, which detects and tracks hand movements in real time. By using hand gesture recognition and fingertip tracking, the AI virtual mouse interprets user commands and translates them into on-screen actions. This technology offers an innovative approach to Human-Computer Interaction (HCI), enabling users to engage with their devices in a way that feels natural and fluid. The system captures hand gestures using a webcam or built-in camera, processes the movements with advanced algorithms, and identifies fingertip positions to simulate mouse actions such as cursor movement, clicks, and scrolling.

One of the key advantages of this AI virtual mouse system is that it eliminates the need for additional hardware, making it a cost-effective and accessible alternative. It enhances usability, particularly for individuals with mobility impairments, by providing a hands-free computing experience. Moreover, it offers a hygienic solution by reducing reliance on physical touch-based devices, which is especially beneficial in shared or public environments. The system also makes computing more interactive and futuristic by allowing users to control their devices with simple hand gestures, reducing dependency on traditional input methods.In conclusion, the AI virtual mouse system represents a significant step toward a more immersive and touch-free computing experience. By harnessing the power of computer vision and AI-driven hand gesture recognition, this technology has the potential to redefine how we interact with our digital devices. Whether for accessibility, convenience, or innovation, gesture-based control is shaping the future of human-computer interaction, paving the way for a more seamless and engaging digital experience.

IMPLEMENTATION

Implementing a virtual mouse using hand gestures and computer vision represents a significant leap forward in how we interact with technology. By eliminating the need for traditional input devices like a physical mouse, this innovative system uses a webcam or built-in camera to detect and interpret hand movements, translating them into mouse actions such as cursor movement, clicking, and scrolling. This approach is not only more intuitive but also provides opportunities for increased accessibility, allowing users to control their devices without relying on physical contact with the computer. The implementation of such a system involves several key steps, ranging from setting up the environment to processing hand gestures and finally mapping these gestures to specific mouse actions.The first step in creating this virtual mouse system is to set up the necessary tools and libraries. Python is an ideal programming language for this project due to its versatility and the wide range of libraries available for computer vision and machine learning. The main libraries used are OpenCV, MediaPipe, and PyAutoGUI. OpenCV is a powerful tool for real-time image processing and handling webcam input, while MediaPipe offers pre-trained models for hand gesture recognition, which makes it much easier to detect and track hand movements. PyAutoGUI is used to simulate mouse actions based on the detected gestures, enabling the system to control the computer's mouse without needing an actual physical device.

Once the environment is set up, the next step is to capture video input from the webcam. This is done using OpenCV, which provides an interface for accessing the camera feed and displaying it in real time. The webcam feed is continuously captured, and each frame is processed to detect hand gestures. In this stage, the goal is to track the user's hand movements and extract relevant information such as the position of the hand and the fingertips. The webcam is essentially acting as the "eyes" of the system, providing the visual data that will later be interpreted by the computer vision algorithms.

With the camera feed in place, the next task is to implement hand gesture recognition. This is where MediaPipe comes into play. MediaPipe offers a highly efficient hand tracking solution, capable of detecting hand landmarks in real time. It uses a deep learning model that identifies the position of key points on the hand, such as the tips of the fingers, the palm center, and the wrist. These key points provide valuable data that can be used to determine the user's gestures. For example, the system can track the movement of a fingertip to simulate the movement of the mouse cursor on the screen. By analyzing the relative positions of these key points, the system can recognize specific gestures, such as a raised finger indicating a mouse click or a swipe gesture for scrolling.

Once hand gestures are detected, the next challenge is to map these gestures to corresponding mouse actions. **PyAutoGUI** is a library that allows for the automation of mouse and keyboard events, making it the perfect tool for simulating mouse actions based on hand gestures. For instance, the system can map the movement of a finger to the movement of the mouse cursor on the screen. When the fingertip moves in a certain direction, the cursor moves accordingly. Similarly, when a user forms a fist or pinches their fingers together, the system can simulate a mouse click. The scrolling function can be triggered by specific gestures, such as moving the fingers up and down or swiping across the screen in a particular pattern.

To achieve this, the positions of the detected hand landmarks are continuously monitored and compared to predefined thresholds or patterns. For example, if the distance between the thumb and index finger decreases, the system interprets it as a click. Similarly, if the hand moves up or down, the system recognizes this as a scroll action. The real-time processing of hand gestures and the mapping of those gestures to mouse actions create a seamless and natural way to interact with the computer without the need for a physical mouse.

One of the primary advantages of this virtual mouse system is that it eliminates the need for additional hardware, such as a wireless mouse or a USB dongle. This can be particularly beneficial in situations where traditional input devices are impractical, such as in presentations, or for users with mobility impairments who find it difficult to use a conventional mouse. The use of hand gestures also reduces the reliance on physical contact with devices, which can be an advantage in environments where hygiene is a concern. For example, in public spaces or during collaborative activities, users can interact with their computers without touching the same physical mouse, reducing the spread of germs.

Moreover, this system introduces a more interactive and immersive form of computing. Traditional input devices like a mouse or keyboard can often feel static and limiting, whereas hand gestures provide a dynamic, fluid way to engage with technology. Users can control the computer by simply moving their hands or making specific gestures, creating a more engaging and futuristic experience. This type of interaction is not only fun but can also be more intuitive, especially when using augmented or virtual reality environments, where gesture-based control is becoming increasingly common.

In terms of technical implementation, real-time processing and accuracy are key challenges in ensuring a smooth user experience. Hand gesture detection models, like the one provided by MediaPipe, need to operate with high precision and minimal latency. This requires efficient image processing and a good balance between computational performance and accuracy. The system must be able to detect and track hand movements in various lighting conditions and from different angles, which can be a difficult task. To address these challenges, it's important to fine-tune the hand tracking model's parameters, such as detection and tracking confidence levels, to optimize performance in real-world scenarios.

Bugs and optimize performance. Common issues, such as background noise affecting voice recognition or lag in response time, are fine-tuned for a seamless user experience.

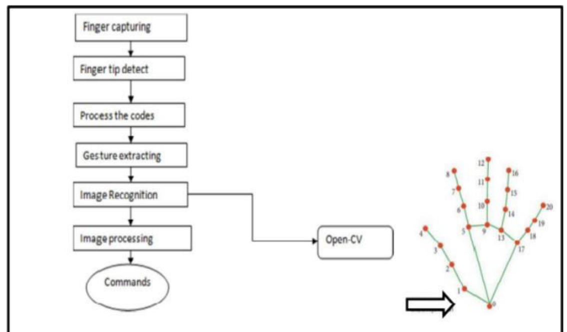


Fig : Flow Chart

Fig- Flow Diagram of virtual mouse

RESULTS

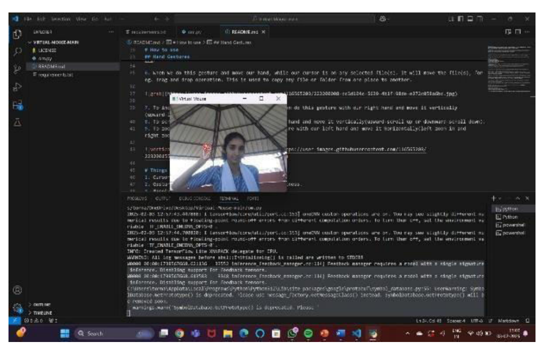
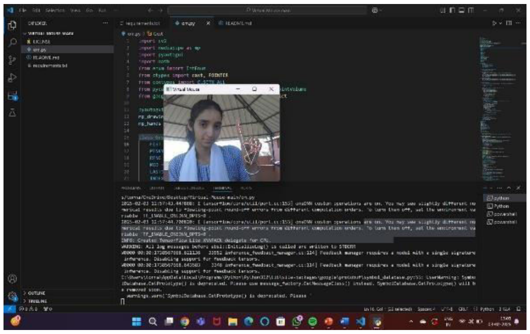


Fig – overview Drag and Drop

Fig – Gesture for single right click



CONCLUSION

The main objective of the AI virtual mouse system is to control the mouse cursor functions by using the hand gestures instead of using a physical mouse. The proposed system can be achieved by using a webcam or a buil in camera which detects the hand gestures and hand tip and processes these frames to perform the particular mouse functions. From the results of the model, we can come to a conclusion that the proposed AI virtual mouse system has performed very well and has a greater accuracy compared to the existing models and also the model overcomes most of the limitations of the existing systems. Since the proposed model has greater accuracy, the AI virtual mouse can be used for real-world applications, and also, it can be used to reduce the spread of COVID-19, since the proposed mouse system can be used virtually using hand gestures without using the traditional physical mouse. The model has some limitations such as small decrease in accuracy in right click mouse function and some difficulties in clicking and dragging to select the text. Hence, we will work next to overcome these limitations by improving the finger tip detection algorithm to produce more accurate results.

REFERENCES

1] Banerjee, A., Ghosh, A., Bharadwaj, K., & Saikia, H. (2014). Mouse control using a web camera based on colour detection. arXiv preprint arXiv:1403.4722..

2]C.-Chiung Hsieh, D.-Hua Liou, & D. Lee, “A real time hand gesture recognition system using motion history image,” Proc. IEEE Int'l Conf. Signal Processing Systems (ICSPS), 2. 10.1109/ICSPS.2010.5555462, 2010.

3]C.-Y. Kao, C.-S. Fahn, "A Human-Machine Interaction Technique: Hand Gesture Recognition Based on Hidden Markov Models with Trajectory of Hand Motion," in Procedia Engineering 15 pp. 3739 – 3743, 2011.

4]P.S. Neethu, “Real Time Static & Dynamic Hand Gesture Recognition,” International Journal of Scientific & Engineering Research, Vol. 4, Issue3, March 2013.- AI Based Personal Desktop," *Int. J. Res. Eng. Sci.*, vol. 9, no. 7, pp. 55–57, Jul. 2021.

5]A. M. Patil1, S. U. Dudhane1, M. B. Gandhi1,” Cursor Control System Using Hand Gesture Recognition”, International journal of advanced research in computer and communication engineering, vol. 2, issue: 5, May 2013.