

AI Based Mouse Controller Using Hand Gestures For Real-Time Applications

"A Deep Learning Approach to Hands-Free Mouse Control"

¹Dr. Bharati Bidikar, ²Gorremuchu Jayasri

¹Adjunct Professor, ²Student

Department of CS&SE, AU College of Engineering

Department of Information Technology and Computer Applications

Andhra University, Visakhapatnam, India

Abstract- *The interaction with the computers by utilizing a revolutionary gesture-based mouse control system has many applications in a variety of sectors, including virtual reality, media management, productivity tools, and gaming. This technology makes use of hand motions tracked by a regular webcam or depth camera to navigate and manage the computer interface, rather than relying on physical mouse devices. The system uses computer vision algorithms to recognize, track, and understand hand gestures in real time in order to execute mouse actions like cursor movement, clicking, and scrolling. The system's key components include hand detection, gesture recognition, and cursor mapping, all of which collaborate to create a seamless and user-friendly interface. The system tackles issues like gesture accuracy and user adaptability via ongoing improvement and feedback integration. Because it provides a more accessible and inclusive mode of interaction, it is particularly helpful for those with motor impairments. This solution, at its heart, illustrates how cutting-edge computer vision and pattern recognition can facilitate smooth and intuitive human-computer interaction, paving the way for intelligent, touchless interfaces.*

Index Terms: Computer vision, hand motion control, gesture-based interface, human-computer interaction, assistive technology.

I. INTRODUCTION

Human-computer interaction (HCI) has evolved into systems that enable users to interact with their digital environment by using simple hand gestures. Using computer vision and machine learning, these gesture-based systems find and react to hand movements in real-time. As it improves accessibility and usability across several applications—including media, productivity, gaming, and assistive technology—controlling a computer mouse with hand gestures has attracted interest. Normally, webcams and computer vision algorithms register and analyze hand movements, therefore producing more precise and interactive models. Building on tried-and-true techniques to increase reliability and flexibility, the present project aims to raise gesture recognition in HCI. Using deep learning and sensor-based tracking for accurate recognition with little delay, the objective is to create a virtual mouse controlled by gestures for several real-world uses. Earlier research have provided useful techniques for real-time gesture recognition.

RESEARCH OBJECTIVES

- To create a real-time mouse interface driven by hand gestures.
- To apply deep learning and computer vision methods for accurate gesture recognition.
- To ensure low-latency gesture-to-command translation using OpenCV, MediaPipe, and PyAutoGUI.

- To design a user interface adaptable to various surroundings and lighting situations for improved accessibility for consumers with limited mobility.

II. LITERATURE REVIEW

The innovative gesture-based mouse control system has multiple applications in various industries, including virtual reality, media management, productivity software, and gaming. Kim and Jung presented a cursor control technique using a joystick that was designed to increase accessibility for users who have restricted movement. Piyush et al. and Shubham et al. created camera-based systems that use OpenCV, MediaPipe, and machine learning to convert hand motions into mouse actions. For gesture-controlled software interaction, Leap Motion systems and CNN-based models were suggested by Samruddhi et al. and Shahed et al. The wearable glove mouse, developed by Vivek et al., uses accelerometers to track hand motions. For media control via gestures, Vaishali et al. used feedforward neural networks. Yu Xiao and Myung-Cheol concentrated on Bayesian networks and CNNs, respectively, for reliable gesture-to-mouse command conversion. The accuracy of identification in a variety of settings was improved by using depth sensors, radar, and RF technologies in subsequent investigations by Heru et al., Ahmed et al., and Palacios et al. Non-contact sign language identification was created by Rahim et al., whereas Cheng et al. suggested multimodal models that combine speech and gesture input. Researchers like Paravati, D'Eusanio, and Malecki created gesture-based interfaces for intelligent environments and automobile systems using RGB-D and infrared sensors. To enhance performance in difficult situations, Chen, Yang, and Moysiadis also investigated sophisticated neural architectures and radar-based solutions. Finally, Nogales et al. employed BiLSTM and ANN models to achieve outstanding accuracy in real-time gesture identification and human-computer interaction.

III. METHODOLOGY

This section discusses the organized methodology used to create a system that can detect real-time hand motions and take the place of conventional mouse devices. Through the use of a webcam or RGB-D sensor, this approach allows for organic user engagement by combining deep learning and computer vision algorithms.

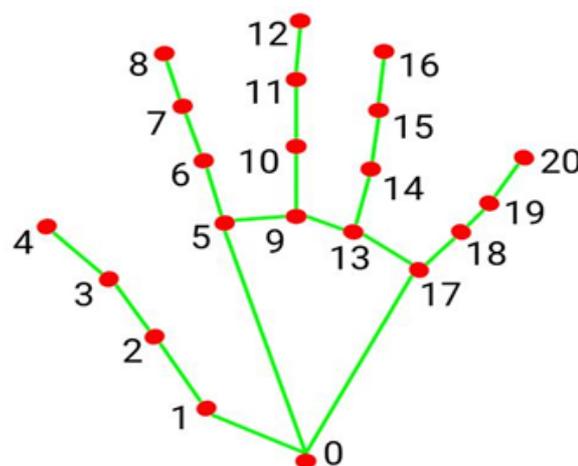


Figure 1: cv detect gestures by tracing these points

Data Acquisition and Pre-processing:

The initial phase consists of grabbing picture frames from a webcam or depth-sensing camera. The precision of hand segmentation is improved by RGB-D data, which includes both color and depth. The environment must have consistent illumination and little background noise for the best detection.

The webcam causes mirror-like inversion in captured images, which is rectified by flipping them horizontally. The last rectified picture is created by independently flipping and reassembling each RGB channel (red, green, blue). The images are then converted to grayscale in order to concentrate on contour-based features for additional processing and lessen the computational load.

Hand Detection

Computer vision techniques like the ones below may be used to detect hands:

- segmentation of skin tone
- classifiers for Haar cascades
- Subtracting the background

These techniques focus hand regions within the frame. The goal is to successfully separate the hand from the background, creating a distinct region of interest (ROI) for gesture detection.

Feature Extraction

Relevant features are extracted after the hand is detected. These are a few of them:

- contour of the hand
- Orientation of the fingers
- Palm Hub
- Coordinates of the fingertips

The accuracy of 3D hand position detection can be improved by using depth data in certain circumstances. The extracted features are used as input for gesture categorization.

Gesture Recognition using Deep Learning

The system employs deep learning models, such as the following, to identify different gestures:

- Spatial feature learning using Convolutional Neural Networks (CNNs)
- Repetitive Neural Networks (RNNs) are used to identify patterns in time.

A labeled dataset of hand gestures is used to train the models. Transfer learning is used to leverage pre-trained models and fine-tune them on the target dataset in situations with a small amount of training data.

Common signals are associated to particular mouse operations, such as:

- cursor motion
- Left-click or right-click
- Scroll
- Drag and drop

The whole setup is made for real-time engagement with low latency. Using automation libraries such PyAutoGUI, gesture inputs are processed frame by frame and converted into mouse control commands at the OS level.

Real-time performance is guaranteed by maximizing the frame processing rate and minimizing the delay between gesture detection and cursor response.

Color Detection and Binary Conversion

A complementary technique that can be used is color-based object tracking. By deducting suppressed color channels from grayscale images, certain colors (red, green, and blue) can be identified. The hand is transformed into a distinct region (white on black background) for simpler processing by converting these images to binary using a set threshold.

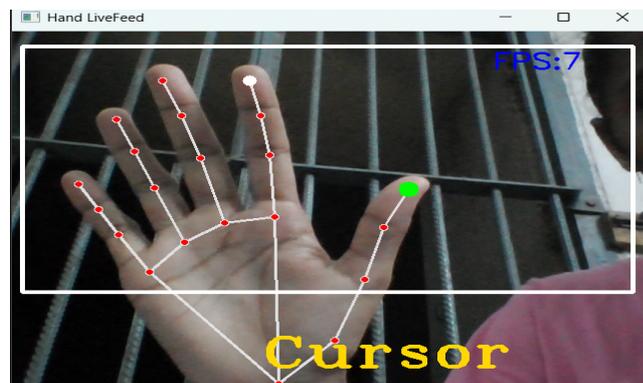


Figure 2: Detection of Hand Gestures

Centroid Detection for Cursor Control

The centroid of the hand or a monitored marker (such as a fingertip) is calculated from the binary image in order to provide precise mouse pointer control. The on-screen cursor is regulated by these coordinates, which are constantly updated and matched to the screen resolution.

User Interface Design

The system has a straightforward interface where users can:

- Change the level of detection sensitivity
- Calibrate movements
- Turn particular controls on or off

Users with various physical capabilities or tastes are ensured inclusivity via customization features.

Implementation Stack

- **OpenCV:** For contour analysis and picture processing
- **MediaPipe (optional):** For identifying hand landmarks
- **PyAutoGUI:** Used to automate the movement of the mouse
- **TensorFlow/Keras:** Used for creating deep learning models

Summary

This approach makes use of simple hand gestures for a contactless and user-friendly way for humans to interact with computers. The system has a futuristic interface for gaming, multimedia, and productivity applications, as well as accessibility for users with impairments.

IV. RESULTS AND DISCUSSION

With its intuitive and natural human-computer interaction, the hand gesture-based mouse control system provides a compelling alternative to traditional input methods. The system is implemented in a development environment like Visual Studio Code, beginning with the capture of hand motions via a regular webcam or a depth-sensing camera. The system first isolates the hand area before extracting key spatial and temporal elements that describe the user's motions, such as finger placement, hand form, movement direction, and gesture trajectory. Machine learning algorithms, notably deep learning models trained on depth data, are used to analyze these retrieved characteristics in order to properly categorize the motions.

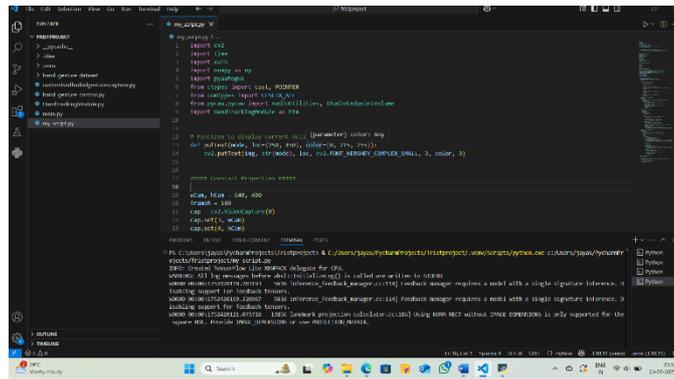


FIGURE 3: Execution of code in Visual studio code

The classified gestures are converted into equivalent mouse operations such as cursor movement, clicking, scrolling, and dragging, enabling the user to operate and navigate the computer system without using a physical mouse. The main goal of the study is to develop a gesture recognition system that is reliable, effective, and able to handle a wide range of scenarios.



FIGURE 4: Output Gesture Recognition

The system uses computer vision technologies and real-time processing to achieve low latency, which leads to a smooth and interactive user experience. The system functioned well during testing in perfect circumstances, correctly identifying both static and dynamic motions and reacting promptly to user input. However, its performance may be impacted by outside influences like fluctuating light and background noise. The system is iteratively improved using user feedback and empirical performance evaluation in order to overcome these issues.

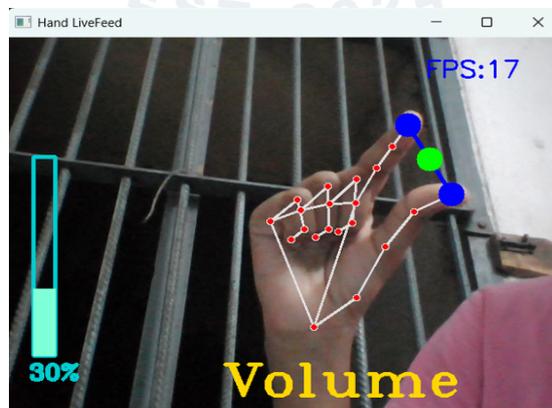


FIGURE 5: Gesture recognition for volume

The system's parameters can be changed and optimized to increase detection reliability and ensure consistent performance in a variety of user contexts. The gesture recognition system not only improves accessibility but also facilitates a more immersive and contactless engagement experience.

The technology has a lot of potential for integration into fields such as robotics, sign language translation, computer graphics, augmented and virtual reality systems, biomedical devices, and even creative platforms like digital canvases utilized by artists, in addition to its direct application as a mouse replacement. Gesture-based interfaces may revolutionize the way users engage with computers in a variety of settings, from daily to professional, as they continue to grow and improve.



FIGURE 6: scrolling gesture recognition

V. CONCLUSION

Gesture recognition technology is a major step forward in the development of human-machine interaction because it allows for a more organic and user-friendly interaction between users and digital systems. Gesture-based control is gradually being incorporated into a number of developing human-computer interaction paradigms as research in this area progresses. The capacity to understand hand gestures and convert them into practical instructions unlocks a vast array of uses, such as immersive environments like augmented reality and virtual reality, sign language translation for people with speech and hearing difficulties, and robotic control. This technology is also essential in areas like biomedical instrumentation, prosthetics, computer graphics, and gaming, where touch-free control is either needed or preferred.

The Digital Canvas and similar platforms in the creative field revolutionize the way digital art is made and experienced by giving artists the power to create two-dimensional and three-dimensional works with just hand motions. The virtual mouse integration enables gestural input without physical interaction, which is particularly helpful for those with restricted mobility, such as limb paralysis. In addition, current gaming systems are increasingly employing gesture-based interfaces to produce engaging and interactive gameplay, in which the player's movements are instantly translated into actions inside the game.

The Mouse Controller Using Hand Gestures initiative has the potential to revolutionize how people use computers, according to its overall conclusion. It expands prospects for creativity in assistive technology, entertainment, and productivity while also improving convenience and accessibility. The system may become a key component in creating more inclusive, responsive, and human-centered computing environments as it continues to develop through iterative improvement and real-world testing.

VI. REFERENCES

- [1] "A Joystick Driven Mouse Controlling Method Using Hand Gestures" by J.-Y. Jung and J.-I. Kim, J. Korea Multimedia Soc., vol. 19, no. 1, pp. 60–67, 2016, doi: 10.9717/KMMS.2016.19.1.060.
- [2] O. Mapari, M. C. Shinde, S. Shigwan, and T. Ambatkar, "Wireless Cursor Controller: Hand Gestures for Computer Navigation," Int. J. Sci. Technol. Eng., vol. 11, no. 3, pp. 62–67, 2023, doi: 10.22214/ijraset.2023.49314.

- [3] "Implementation of Hand Gesture-Controlled Mouse Using Artificial Intelligence," by P. Pilare, C. Mahato, C. Khergade, S. Agrawal, and P. Thakre, vol. 11, no. 2, pp. 71–79, 2022, doi: 10.17993/3ctecno.2022.v11n2e42.71-79.
- [4] S. Mumbare et al., "Software Controller Using Hand Gestures," ITM Web of Conf., vol. 50, p. 01008, 2022, doi: 10.1051/itmconf/20225001008.
- [5] "Hand Swifter: A Real-Time Computer Controlling System Using Hand Gestures," by S. Anzarus, S. Islam, M. Hossain, and S. Shahreen, 2018, doi: 10.1109/CEEICT.2018.8628099.
- [6] V. K. Singh et al., "Glove Mouse: A Gesture Controlled Wireless Computer Mouse," in Proc. Int. Conf., pp. 483-494, 2021, doi: 10.1007/978-981-16-0336-5_40.
- [7] "Mouse Navigation Control Using Hand Gesture with Neural Network" by V. M. Gulhane and M. S. Joshi, 2013.
- [8] Int. J. Sci. Technol. Eng., vol. 11, no. 5, pp. 6642–6646, 2023, doi: 10.22214/ijraset.2023.52881, "Computer Vision Based Media Control Using Hand Gestures".
- [9] Y. Xiao and Q. Peng, "A Hand Gesture-Based Interface for Design Review Using Leap Motion Controller," pp. 239–248, 2017.
- [10] "Controlling Media Player Using Hand Gestures," vol. 2, no. 1, pp. 45–54, 2023, doi: 10.46632/ese/2/1/7.

