EasyChair Preprint

№ 12652

# Gesture Controlled Mouse Using Python

Bhavika Darji, Akula Mahesh, Akula Tharun, Akula Harsha and Alladi Bharadwaj



March 21, 2024

# GESTURE CONTROLLED MOUSE USING PYTHON

*Bhavika Darji*
*Assistant Professor*
*CSE Department Parul University*
Vadodara, India
bhavikaben.darji19384@paruluniversity.ac.in

A Mahesh
*CSE AI Parul University*
Vadodara, India
200303124105@paruluniversity.ac.in

A Tharun
*CSE AI Parul University*
Vadodara, India
200303124106@paruluniversity.ac.in

A Harsha Vardhan
*CSE AI Parul University*
Vadodara, India
2003030124104@paruluniversity.ac.in

A Bharadwaj
*CSE AI Parul University*
Vadodara, India
2003030124108@paruluniversity.ac.in

***Abstract*—The technique of establishing a process of interaction between human and computer is evolving since the invention of computer technology. In terms of HCI (Human-Computer Interaction) technology, the mouse is a fantastic invention. Even while Bluetooth and wireless mouse technology is still in its infancy, it is not entirely device independent. A Bluetooth mouse needs a connected dongle and batteries to function. When a mouse has additional devices, it becomes harder to use**

**The proposed mouse system is beyond this limitation This project proposes a gesture-controlled mouse using Python, where the user can control the movement of the cursor on the computer screen with hand gestures. The system uses a webcam to capture real-time video input and OpenCV, a computer vision library in Python, to detect and recognize hand gestures. The hand gestures are mapped to the movement of the cursor on the screen using PyAutoGUI, a Python library that allows for programmatically controlling the mouse and keyboard. The proposed system aims to provide an intuitive and hands-free approach to computer interaction, especially for users with physical disabilities or for those who prefer a more natural way of controlling their computer. The system will be implemented and tested on a Windows platform using Python 3.x, OpenCV 4.x, and PyAutoGUI. The performance and usability of the system will be evaluated through a user study, where participants will be asked to perform various tasks using the gesture-controlled mouse and provide feedback on their experience. Overall, this project aims to demonstrate the potential of using hand gestures as an alternative input method for controlling a computer mouse and to provide a useful tool for people who have difficulty using traditional input devices**

## I. INTRODUCTION

The traditional way of interacting with a computer involves the use of a mouse and keyboard, which can be challenging for some users, such as those with physical disabilities. A gesture-controlled mouse system can provide an intuitive and hands-free approach to computer interaction, making it accessible for everyone. This project proposes a gesture-controlled mouse using Python, which allows users to control using hand motions to move the pointer on a computer screen

The system detects and recognizes hand motions using OpenCV, a Python computer vision package, and records live video input using a webcam. PythonAutoGUI is a package that enables programmatic control of the mouse and keyboard. It maps hand movements to the movement of the cursor on the screen

The proposed system aims to provide a natural and effortless way of controlling a computer, especially for users with physical disabilities. The Windows platform will be used for the system's implementation and testing. on a Windows platform using Python 3.x, OpenCV 4.x, and PyAutoGUI. The performance and usability of the system will be evaluated through a user study, where participants will be asked to perform various tasks using the gesture-controlled mouse and provide feedback on their experience.

The overall goals of this project are to show off the possibilities of hand gestures as a substitute input technique for a computer mouse and to offer a helpful tool to individuals who have trouble with conventional input methods. A thorough explanation of the suggested system, including its hardware and software components and the testing and implementation process, will be given in the following section of the article

## II. LITERATURE SURVEY

Gesture recognition and control have garnered significant attention due to their potential applications in various fields, including human-computer interaction (HCI) and assistive technology. In recent years, researchers have explored different approaches and techniques to develop gesture-controlled systems, particularly focusing on Python-based implementations for their versatility

Python's popularity and extensive libraries make it a preferred choice for developing gesture-controlled systems. The availability of libraries such as OpenCV, TensorFlow, and PyTorch streamlines the implementation of various gesture recognition techniques. OpenCV, a widely used computer vision library, provides robust unctionalities for image processing, feature extraction, and object tracking, essential for gesture recognition tasks. Its intuitive Python interface facilitates rapid prototyping and experimentation, enabling researchers to develop gesture-based interfaces with minimal effort.TensorFlow and PyTorch, leading deep learning frameworks, empower researchers to build complex neural network architectures for gesture recognition. Their high-level APIs and extensive documentation simplify model development and training, fostering innovation in gesture-controlled applications

## III. SCOPE AND METHADOLOGY

### *A. Aim of the project*

The project aims to demonstrate the feasibility and effectiveness of using hand gestures as an alternative input modality for controlling computers or devices, with a specific focus on Python-based implementation and real-world usability

### *B. Object of the system*

**Enhanced Accessibility:** Enable individuals with physical disabilities or limitations to interact with computers more effectively by offering a gesture-based input modality that does not require fine motor skills or dexterity

**Improved User Experience:** Provide users with a novel and engaging interaction experience that leverages intuitive hand gestures, potentially reducing cognitive load and enhancing user satisfaction compared to conventional input methods

**Efficient Gesture Recognition:** Develop robust and real-time gesture recognition algorithms capable of accurately interpreting a wide range of hand gestures, including gestures for cursor movement, clicking, dragging, scrolling, and other common mouse actions

### *C. Proposed system*

**Image Upload:** Users can upload an image containing a hand gesture through a graphical user interface.

**Gesture Detection:** Upon image upload, the system uses the HandTrackingModule to detect the hand gesture present in the image.

**Gesture Annotation:** After detecting the gesture, the system prompts the user to input an action corresponding to the detected gesture.

**Data Storage:** The system stores the detected gesture-action mapping in an Excel file

**Error Handling:** The system includes error handling mechanisms to catch and print any exceptions that occur during the gesture detection process

Overall, the proposed system serves as a tool for collecting and annotating hand gesture data, which could be useful for training and testing gesture recognition models or applications. Users can iteratively upload images, annotate gestures with corresponding actions, and accumulate a dataset for further analysis or development purposes

### *D. Scope of the system*

The proposed system focuses on image-based gesture recognition and annotation, encompassing several key functionalities. It utilizes computer vision techniques, particularly hand tracking, to detect and track hand gestures within uploaded images. Through a user-friendly graphical interface developed using tkinter, users can interact with the system by uploading images containing hand gestures and providing corresponding actions for those gestures. The system facilitates the annotation of detected gestures by prompting users to input actions associated with each recognized gesture, enabling the creation of labeled datasets crucial for training and evaluating gesture recognition algorithms

Furthermore, the system manages gesture-action mappings, serving as a centralized repository for storing annotated data. It incorporates error handling mechanisms to gracefully handle exceptions during gesture detection or data processing, ensuring smooth user experience. Additionally, while the current implementation focuses on single-image gesture annotation, the system offers extensibility by potentially supporting batch processing of images, real-time gesture detection from video streams, or integration with other data formats and storage mechanisms. The proposed system holds integration potential in various contexts, including gesture recognition research, development of gesture-based applications, and human-computer interaction studies. By providing a foundational tool for collecting and managing gesture data, it facilitates further research and development in the field of gesture-based interaction and computer vision.

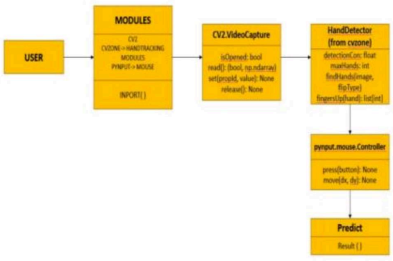


Fig 3.1 Flow Diagram

## IV. IMPLEMENTATION

Key components and tools include:

**Tkinter Library:** Tkinter is utilized for developing the graphical user interface (GUI) of the system. As the standard GUI toolkit for Python, Tkinter provides the necessary widgets and functionality for creating interactive interfaces, allowing users to upload images and provide corresponding gesture actions

**OpenCV (Open Source Computer Vision Library):** OpenCV is employed for image processing and hand tracking within the uploaded images. This library offers a wide range of functionalities for tasks such as object detection, feature extraction, and image manipulation, making it instrumental in detecting and tracking hand gestures

**cvzone Library:** The cvzone library, specifically its HandTrackingModule, is utilized for efficient hand detection and tracking within the images. This module simplifies the process of detecting hand landmarks and recognizing gestures, enhancing the system's accuracy and performance

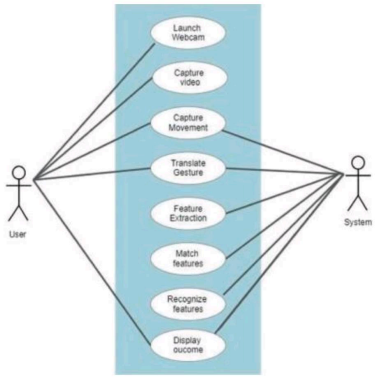


Fig 4.1: Use Case Diagram

**Implementation Workflow:**
**Image Upload:** Users interact with the system through the Tkinter-based GUI, where they can upload images containing hand gestures using the provided interface
**Gesture Detection:** Upon image upload, the system utilizes OpenCV and the cvzone HandTrackingModule to detect and track hand gestures within the uploaded images. The HandTrackingModule identifies key hand landmarks and recognizes gestures based on their configurations.
**Gesture Annotation:** Following gesture detection, users are prompted to input actions corresponding to the recognized gestures using the GUI. This annotation process allows users to specify the intended actions associated with each detected gesture
**Data Management:** Annotated gesture-action mappings are managed using the Pandas library, where they are stored in a structured format. The data is stored in memory during the session and can be saved to a file for future reference or analysis

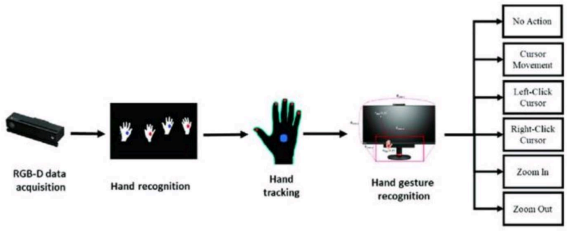


Fig 4.2 Model Diagram

Fig 4.3 Implementation

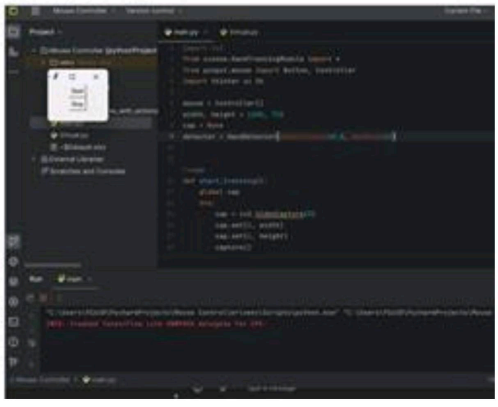

Fig 4.4 Implementation

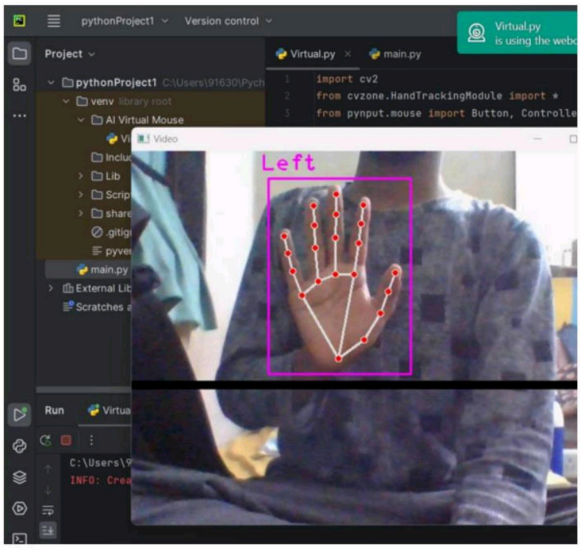


Fig 4.5 Result

## V. CONCLUSION

The suggested system's aim and goal is to take control of the system without utilizing wired or wireless control methods. With the help of hand gestures, this technology enables us to operate the mouse, which in turn can control the system. This is accomplished by giving the user access to a camera that is integrated into the system and which manipulates hand movements and carries out associated tasks

Creating a gesture-controlled mouse using Python involves leveraging computer vision libraries, selecting appropriate hardware like webcams, defining and mapping gestures, ensuring a smooth user experience, and addressing challenges like lighting conditions. Continuous testing and documentation are crucial for refining the system and making it user-friendly and shareable

Implementation of the proposed system leverages Python and various libraries to create an interactive tool for image-based gesture recognition and annotation. By combining computer vision techniques with user-friendly interface design, the system enables efficient gesture annotation, fostering the creation of labeled datasets for training and evaluating gesture recognition algorithms. With its modular architecture and extensible design, the system can be further enhanced and integrated into larger projects or workflows involving gesture-based interaction and human-computer interaction research

## VI. **REFERENCES**


1. D.L. Quam, “Gesture recognition with a Data Glove,” 10.1109/NAECON.1990.
2. Abhik Banerjee, Abhirup Ghosh, Koustuvmoni Bharadwaj,” Mouse Control using a Web Camera based on Color Detection”,IJCTT, Mar 2014.
3. Angel, Neethu.P.S,”Real Time Static Dynamic Hand Gesture Recognition”, International Journal of Scientific Engineering Research March-2013.
4. Game P. M., Mahajan A.R,”A gestural user interface to Interact with computer system ”, International Journal on Science and Technology (IJSAT) (Jan.- Mar.) 2011.
5. Oudah, Munir, Ali Al-Naji, and Javaan Chahl. ”Hand gesture recognition based on computer vision: a review of techniques.” journal of Imaging 6, no. 8 (2020): 73.
6. Guoli Wang, (2010). Optical Mouse Sensor-Based Laser Spot Tracking for HCI Input, Proceedings of the 2015 Chinese Intelligent Systems
7. Kaoru Y, Lie J, Cheng Z (2014) System for controlling personal computers by hand gestures using a wireless sensor device. IEEE