Controlling laptop working with hand gestures

|  |  |  |  |
| --- | --- | --- | --- |
| Kaustubh Naithani 20BCS4000  [20bcs4000@cuchd.in](mailto:20bcs4000@cuchd.in)  Chandigarh University  Mohali,Punjab India | Dhruva Malik 20BCS3976 [20bcs3976@cuchd.in](mailto:20bcs3976@cuchd.in)  Chandigarh UniversityMohali,Punjab India | Prashant Kumar  20BCS4363  [20bcs4363@cuchd.in](mailto:20bcs4363@cuchd.in)  Chandigarh University  Mohali,Punjab  India | Mohammed Bilal  20BCS3994  [20bcs3994@cuchd.in](mailto:20bcs3994@cuchd.in)  Chandigarh University  Mohali,Punjab  India |

Dr Monika Singh

Chandigarh University

Mohali,Punjab

India

Abstract

In Human-Computer Interaction (HCI), the traditional mouse is a remarkable invention in computer technology which encounters limitations in modern Bluetooth and wireless models due to battery dependence and PC connectivity via dongles. These challenges are tackled by the proposed AI-driven hand gesture-based virtual mouse. By capturing hand movements via a webcam, this innovation employs gestures as a potent communication mode. It allows the computer to replicate functions like left and right-click, scrolling, and cursor control without a physical mouse, aiding in COVID-19 prevention and reducing bias in computer control. This system is integrated with voice commands and utilizing Machine learning (ML) and Computer Vision algorithms. It eliminates extra hardware requirements, enhancing accessibility and user interaction in diverse environments. Integrating new HCI technologies might be difficult because of software support requirements and compatibility problems. However, by providing more natural and intuitive interactions, lowering the learning curve, and enabling cutting-edge capabilities like gesture controls and multi-device connectivity, they can improve productivity and the user experience while eventually speeding up and streamlining activities.

Keywords :- Human Computer Interaction (HCI), Webcam, OpenCV, Mediapipe , Machine Learning, Virtual Mouse.

# Introduction

Computers have seamlessly integrated into our daily routines, simplifying human-computer interaction (HCI). Important developments in HCI have helped modern mouse models, which together improve user experiences. Battery life is increased by energy-efficient parts like low-power sensors and CPUs. Wireless charging makes charging convenient by doing away with the requirement for throwaway batteries. Power conservation and great precision are guaranteed by advanced sensor technology. Users can move between devices with ease thanks to multi-device connectivity, which lessens the need for conventional dongles. All things considered, universal receivers make connecting even easier, which lowers the need for batteries and improves user convenience.

New HCI technology can address some battery and connectivity issues found in traditional mouse devices. To address battery concerns, for example, it can include wireless charging and energy-efficient components. Furthermore, more sophisticated wireless technologies like Bluetooth Low Energy can increase the dependability of connectivity. These fixes, however, would not fully resolve every problem and might even create new difficulties unique to the technology in use.

However, individuals with disabilities face distinct challenges in utilizing these devices effectively. It introduces a hand gesture-based AI virtual mouse system, leveraging computer vision to execute mouse tasks using hand movements through mediapipe hand landmark detection technique. The system's core objective is to replace the conventional mouse with a web or built-in camera, enabling cursor control and scrolling through hand gestures.

In terms of power efficiency, conventional mice, which are generally wired, draw power directly from the computer via the connection port. Wireless mice, on the other hand, use batteries that need to be replaced or recharged periodically. Modern Human-Computer Interaction (HCI) devices, such as touchscreens and gesture-based systems, are typically integrated into the device like a smartphone or tablet and share the device’s power source. The power efficiency of these devices varies based on the specific technology and usage.

When it comes to PC connection, conventional mice connect to the PC via a USB port, PS/2 port, or wirelessly via Bluetooth. Modern HCI devices use a variety of connection methods. Touchscreens are typically integrated into the device, while other technologies may use USB, Bluetooth, or other wireless technologies. The specifics can vary based on the exact models and technologies used.Python, along with the OpenCV computer vision library, forms the foundation of the hand gesture-based AI virtual mouse framework. MediaPipe aids in tracking hand and fingertip positions, while Autopy facilitates window navigation, left and right clicking, and scrolling actions. Notably, the model demonstrates exceptional accuracy and efficiency, particularly on CPU-based systems, extending its practicality to real-world scenarios.Hand gestures are universally recognized as a potent form of human expression. The system taps into this natural mode of communication, proposing a low-cost USB web camera setup for input. The proposed system introduces a real-time hand gesture system, incorporating techniques such as preprocessing, background subtraction, skin detection and edge detection to achieve effective gesture segmentation. By employing the Python language and OpenCV, the AI virtual mouse system is crafted, further enhanced by MediaPipe, Autopy, and PyAutoGUI packages for precise hand tracking, cursor movement, and diverse operations.It presents a comprehensive approach to bridging accessibility gaps through a hand gesture-driven AI virtual mouse, ushering in a new era of intuitive human-computer interaction (HCI) [1].

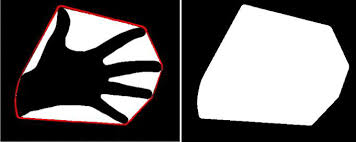
# LITERATURE SURVEY

**Existing System**

The Virtual Mouse Control System, employing colored fingertip detection and hand gesture recognition, enables cursor manipulation without requiring direct physical contact or sensors. This approach involves identifying vibrant fingertip colors and tracking their movements, with the potential to substitute different hand gestures for colored caps. The system facilitates a range of mouse functions, including scrolling, single and double-clicking on the left side, and more. Diverse arrangements of colored caps cater to various tasks, adapting to different users and lighting conditions. The system's adaptability is demonstrated by adjusting skin tone detection based on the user and environment. By analyzing program output during hand motions, the unused area within the hand's convex hull [2] is approximated. The exiting system utilizes the convex hull defects to determine hand gestures. In well-lit settings like offices, where brightness ranges from 500 to 600 lux, colors like Red, Green, and Blue exhibit around 90% detection accuracy. The system overcomes these challenges by incorporating hand gesture recognition technology that detects hand contours, effectively enabling mouse operations through colored fingertips.

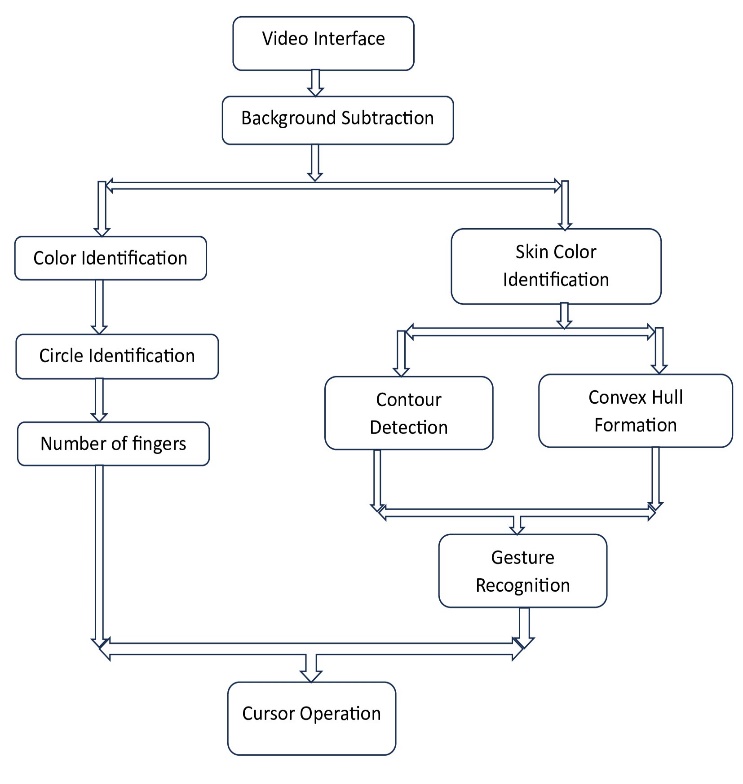


**Fig 1: -** Detected contour of hand gesture



**Fig 2 : -** Convex hull of hand gesture

This system comprises two main methods: finger detection via colored fingertips and gesture recognition. It employs video processing and background subtraction, leveraging colored finger caps for detection, followed by color and circle identification [3]. The recognized gestures trigger various mouse actions[4].



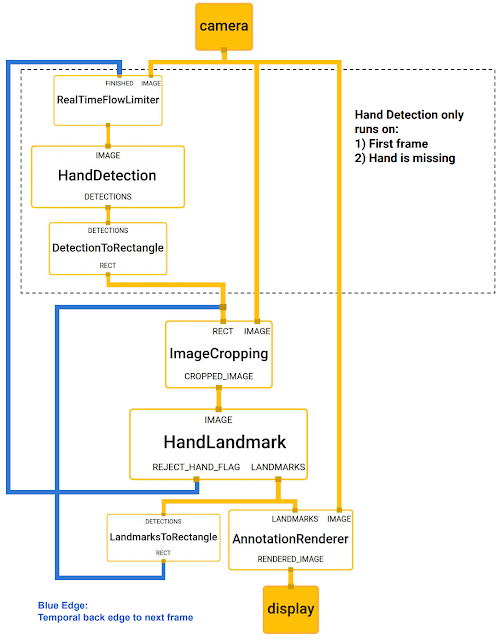
## **Fig 3 : -** Virtual Mouse Control using Colored Fingertips and Hand Gestures Recognition

**Proposed System**

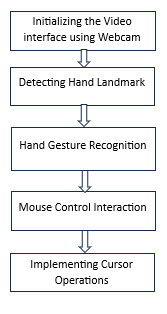
Creating a hardware-independent virtual mouse eliminates the need for additional equipment and colored fingertip markers. Instead, our approach utilizes the existing web camera on the user's device to capture real-time feeds. The system's workflow commences by initiating the video interface through the device's web camera. Subsequently, the system detects hand gestures, facilitating mouse interaction and executing diverse cursor operations [5][6]. This virtual mouse solution is characterized by its portability, user-friendliness, and affordability. In the proposed systems, user’s are not required to use any sensors or gadgets.

In the proposed system, the OpenCV library is used for computer vision tasks and the Mediapipe framework is used for hand tracking and gesture recognition. Especially it is used for hand landmark detection. Mediapipe framework uses machine learning algorithm to differentiate between the hand gestures and fingers of the detected hands. Mediapipe marks 21 hand knuckle coordinates within the detected hand regions through machine learning. This integration enhances the virtual mouse's capabilities and precision, contributing to an improved user experience [7][8].

Mediapipe graph for hand tracking is shown below. The graph consists of two subgraph – one for hand detection and other for hand landmark detection.



**Fig 4 : -** Mediapipe Graph



**Fig 5 : -** Virtual mouse using hand gesture recognition

**Table:-** Comparison between the existing system and proposed system

|  |  |  |
| --- | --- | --- |
| **Factor** | **Existing System** | **Proposed System** |
| Complexity | Complex, requires custom development | Relatively straightforward API |
| User Base | Limited, specific hardware and object | Broader potential user base |
| Merits | Unique, customizable interaction method | Versatile, well-supported framework |
| Demerits | Sensitive to lighting and color changes | Internet connection for setup |
| Processing Requirements | Varies based on implementation complexity | May have higher resource requirements |
| Robustness to Environmental Factors | Sensitive to lighting and background | Robust to changes in lighting |
| Gesture Recognition Capabilities | Limited, often basic gestures | Pre-trained models for various gestures |
| Real-time Feedback | May be challenging to implement | Supports real-time feedback |
| Calibration and Setup | Requires calibration and setup steps | Straightforward setup process |
| Community and Support | Limited community support | Strong community and ongoing updates |
| Integration into Other Technologies | Requires more custom development | Designed for integration with technologies |

# Methodology

**OpenCV :**

OpenCV is a Python-based computer vision library which encompasses object detection and image processing algorithms. It's integral for real-time computer vision applications, image/video processing, and tasks like face and object detection.

**Mediapipe :**

MediaPipe is an open-source Google framework which serves as a versatile tool for machine learning pipelines, particularly for cross-platform development due to its time series data compatibility and multimodal capabilities.

**PyAutoGUI :**

PyAutoGUI is a versatile Python module which facilitates a cross-platform GUI automation for users. It enables mouse and keyboard control in Python scripts and allowing automation of interactions with various applications and simplifying repetitive tasks.

**Algorithm:-**

Step 1: Start

Step 2: Start the webcam video capture and initialize the system.

Step 3: Frame of the video is capture with a webcam.

Step 4: To detects hand and finger tips used mediapipe and OpenCV.

Step 5: Draw hand landmarks on the detected hands.

Step 6: Determine which finger of the hands is raised.

Step 7: If all the five fingers of the hand are up then the gesture is neutral and again proceed to step 2.

Step 8: When only the index finger is up then cursor movement is performed and again proceed to step 2.

Step 9: If middle finger is down then the left click is performed and again proceed to step 2.

Step 10: If the thumb and index finger touches then scroll up is performed and again proceed to step 2.

Step 11: If thumb and middle finger touches then scroll down is performed and again proceed to step 2.

Step 12: If the thumb and pinky finger is up then sound decreases and again proceed to step 2.

Step 13: If the thumb, index finger and pinky finger is up then sound increases and again proceed to step 2.

Step 14: If you make a hand fist then the whole application closed.

Step 15: Exit.

The above algorithm shows how to simulate the controlling action of virtual mouse with hand gestures step by step.

Following are approaches used in the proposed system :-

**Capturing the video :-**

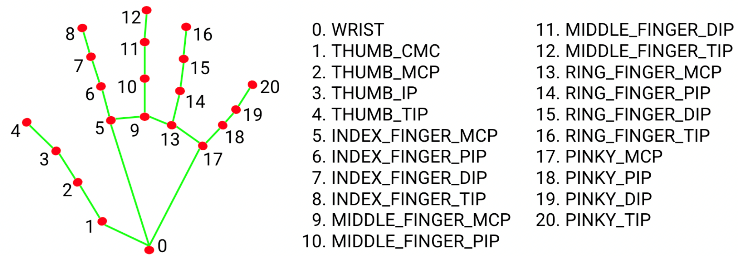
This AI virtual mouse system utilizes the Python computer vision tool OpenCV to create a video capture object. A web camera records frames, subsequently processed by the virtual AI system.

**Analysing the video for hand gesture :-**

The AI virtual mouse system employs a webcam to capture frames throughout the program execution. These images are converted to RGB, facilitating hand identification on a frame-by-frame basis.

**Hand recognising Landmarks :-**

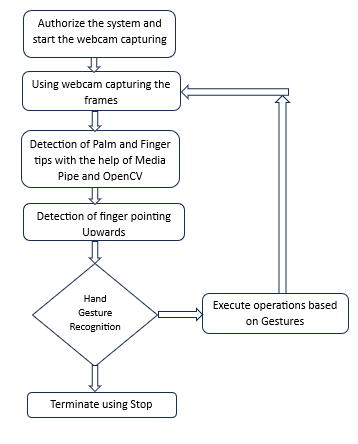
After analyzing hand we flipped the image into gray scale image and then convert it into binary scale image. After that we use mediapipe to detect the landmarks of the detected hands. For the user to control the mouse it is necessary to determine the points whose coordinate can be send to the cursor.



**Fig 6:-** Landmarks in Hand

**Checking the finger which is up and performing mouse operation :-**

For seamless hand coordinate transfer from webcam to full-screen computer window, the AI virtual mouse employs a transformative approach. Once hands are detected, specific finger movements recognized, and coordinate determined. This allows easy mouse pointer movement within the window according to the recognized hand gestures.



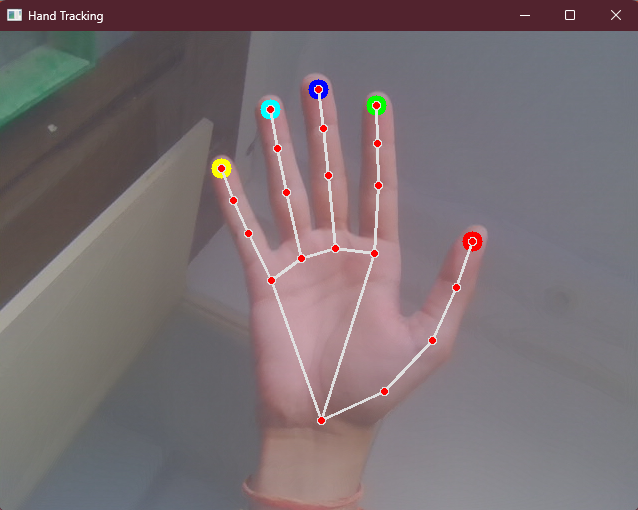
**Fig 7 :-** The real-time gesture-based virtual mouse system's flowchart

# Results

For testing the implementation on a personal computer, the various hand gestures recognized by media pipe hand landmark detection technique and PyautoGUI can be used to control the computer mouse has been shown.

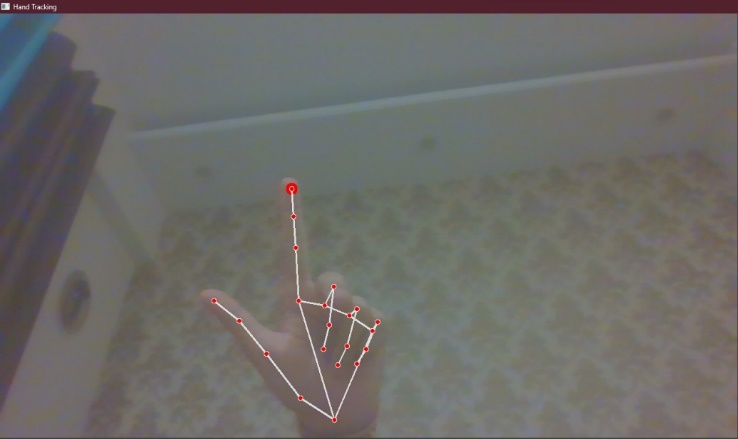
**Table :-** The number of times the project model correctly identified the performed actions

|  |  |  |
| --- | --- | --- |
| Actions | Number of times performed | Number of times it was accurately executed |
| Cursor Movement | 50 | 49 |
| Mouse Button Clicks | 50 | 44 |
| Scroll up | 50 | 47 |
| Scroll down | 50 | 46 |
| Zoom in | 50 | 48 |
| Zoom out | 50 | 44 |
| Volume up | 50 | 45 |
| Volume down | 50 | 46 |
| Closing | 50 | 48 |



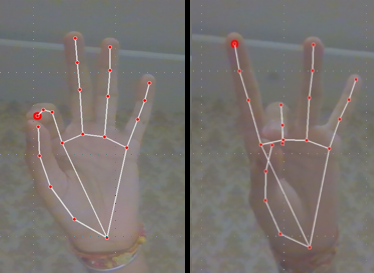
**Fig 8 :-** Open hand

This shows the open hand with colored fingertips which makes it easier to differentiate different fingertips



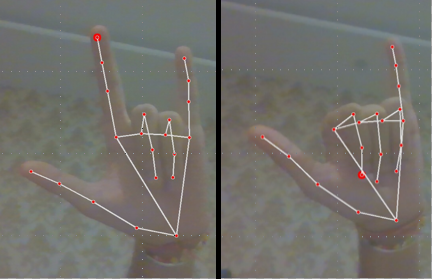
**Fig 9 :-** Bending middle finger to perform Clicks

Middle finger when folded, triggers the left click function of the mouse



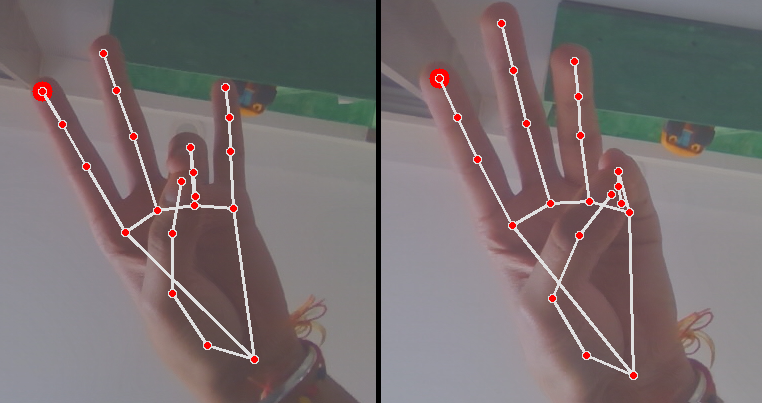
**Fig 10 :-** Scroll Up and Scroll Down

If index finger is touched to the thumb, scroll up command is executed, whereas if the middle finger is touched to the thumb, scroll down is executed.



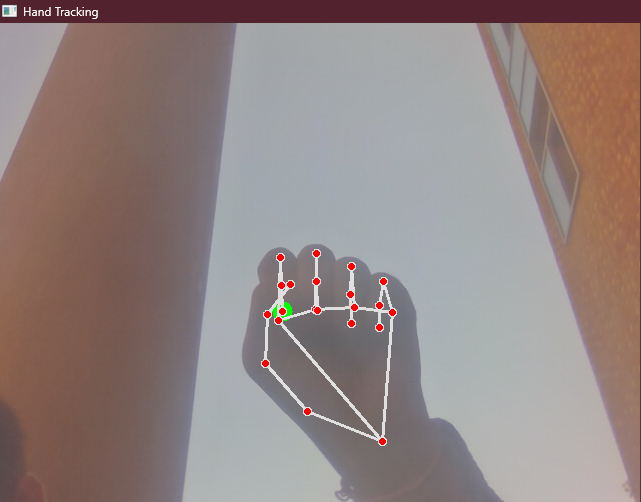
**Fig 11 :-** Changing volume

There are many times when a person needs to change volume. If the thumb, index finger and pinky is open, volume up command is executed, whereas if only the thumb and pinky are open, volume down is executed.



**Fig 12 :-** Zooming in and out

There are many times when a person needs to change the sizes of the text or the web page, due to bad eyesight or distance from the screen. If index ring finger is touched to the thumb, zoom in command is executed, whereas if the pinky finger is touched to the thumb, zoom out is executed.



**Fig 13 :-** closed fist

When there is need to close the application after usage, You can form a closed fist , and that kills and stops the process.

# Conclusions

In conclusions, the development of a virtual mouse through hand gestures represents an exciting frontier in Human Computer interaction (HCI). Leveraging computer vision and machine learning, real-time hand gesture detection and mapping to mouse functions using PyAutoGUI opens up innovative possibilities for users to control applications intuitively. The integration of Mediapipe has streamlined and enhanced hand tracking. This technology has the potential to transform computer interaction into a more natural and instinctive experience. With further refinement, virtual mouse control with hand gestures may become a widely adopted technology across various applications, from gaming to productivity tools.

# Future Scope

The future of virtual mouse technology holds significant promise in enhancing PC understanding and achieving diverse objectives. Despite current limitations in right-click precision and text selection, ongoing research aims to address these issues. Additionally, incorporating keyboard functionality for simultaneous keyboard and mouse operations is a prospective advancement. Future developments may expand gesture options, allowing users to efficiently accomplish a broader range of tasks. Potential improvements include utilizing both hands for various gestures, opening doors to enhanced applications through rapid advancements in hand gesture recognition systems.

##### References

1. Gitte, R., Kharade, R., Mundhe, P., Patil, M. and Patil, R.A., HAND GESTURE RECOGNITION AND CURSOR CONTROL.
2. Hand gesture recognition based on convex defect detection

https://www.ripublication.com/ijaer17/ijaerv12n18\_04.pdf

1. VIRTUAL MOUSE USING OPENCV (volume 10), 2023

https://www.irjet.net/archives/V10/i3/IRJET-V10I358.pdf

1. Kharche, O., Kad, S., Rathod, S. and VS, S., GESTURE MOUSE CURSOR CONTROL USING OPEN CV.
2. Sankar, M.E., Bharadwaj, B.N. and Vignesh, A.V., 2023. Virtual Mouse Using Hand Gesture.
3. Srinivas, G.N., Pratap, S.S., Nagapriya, K.G. and Rao, A.V.S., 2023. Virtual Mouse Control Using Hand Gesture Recognition.
4. Malathi, C., Madhuri, K., Anjali, B., Sathish, M. and Krishna, M.J., Virtual Mouse using Hand Gestures and Landmarks.
5. Latha, B., Sowndarya, S., Swethamalya, K., Raghuwanshi, A., Feruza, R. and Kumar, G.S., 2023. Hand Gesture and Voice Assistants. In *E3S Web of Conferences* (Vol. 399, p. 04050). EDP Sciences.