



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

Contactless Gesticulation Control System using Opencv & Mediapipe

Ritik Raushan, G. Nivetha Sri, Sachin Kumar Yadav, Samji Sai Rohith, Sashang Umanath S

Dept. of CSE, Bharath Institute of Higher Education and Research, Chennai, India

ABSTRACT: The Contactless Gesture Navigation System offers a new way to control computers, using hand gestures instead of mouse, recognizing movements instantly. Using smart camera tech and a basic webcam, it moves the cursor, clicks, and scrolls without touch. This tech is ideal for clean places like hospitals and public areas, where touching is risky. You can customize gestures for natural, quick control. This touch-free tech is safer, easier, and works anywhere.

KEYWORDS: Gesture Recognition, Contactless Navigation, Computer Vision, Human-Computer Interaction (HCI, Hand Tracking, Touchless Interaction,), Real-Time Processing, Customizable Gestures.

I. INTRODUCTION

Mechanical input devices such as mice and touchpads require direct contact, which sometimes is uncomfortable or unhygienic, especially in public and medical environments. This increased demand for touchless interaction has led to the development of gesture-based control systems, based on computer vision and artificial intelligence, to provide a simple input facility compared to the conventional input activities. Observing hand gestures, such systems can perform substantial navigation tasks such as cursor movement, clicking, and scrolling without direct physical touch.[1]

The Contactless Gesture Navigation System employed in the project utilizes a webcam and machine learning-based hand-tracking techniques for real-time user tracking [2]. This supports smooth and precise cursor control and is extremely customizable in different lighting conditions and hand shapes. The gesture command personalization also provides user-specific functionality, which supports enhanced usability and productivity in different applications. The flexibility in deployment supports users from diverse backgrounds to naturally interact with digital interfaces, thus improving accessibility [3,4].

The new technology aims at bridging the gap between gesture-based navigation and secure authentication through a contactless, intuitive, and highly responsive substitute for traditional input devices. It encourages usability, security, and convenience and can be used in personal as well as business environments [5]. Future research can explore the integration of AI-based adaptive learning to further refine the fine-tuning of gesture recognition and further improve interaction with virtual and augmented reality systems [6]. Current research indicates that AI-based adaptive learning enhances the accuracy of gesture recognition and supports high degrees of personalization, with a more customized experience in dynamic settings [7]. In addition, the integration of deep learning models with real-time tracking can further enhance the responsiveness and accuracy of gesture-based navigation systems. Research in the field has also indicated the need to enhance computational efficiency at the expense of gesture recognition on low-resource devices to enable extension of applicability to other technology platforms [8].

II. GESTURE CONTROL SYSTEM

The focus of our system is to provide the user an intuitive, natural, gesture-based interaction methods through the replacement of traditional devices by real-time gesture recognition.

Proposed System: The system will help in a smooth cursor movement, clicking, and scrolling with hands gestures combination with level of understanding butterfly movement, clicking, and scrolling with hand gestures combination with security.

It overcomes the limitations of conventional devices, which requires physical contact and is more safe than traditional devices. This system grip the power of advanced computer vision and machine learning-based hand tracking technologies to provide accurate and real-time gesture recognition. In addition to provide accurate and real-time gesture recognition. In addition to security, it adds security and blocks unauthorized access. The end goad it to develop an extremely flexible and user-friendly system for improving accessibility, usability and efficiency. This project aims to close the gap between gesture-based navigation and secure authentication and allows a futuristic, contactless. and intelligent alternative to conventional input devices.

III. IMPLEMENTATION

This project aims to replace conventional input devices like mouse or touchpad eliminating the element of touch for cursor control. This is achieved by first capturing hand movements via a commodity webcam , followed by interpreting it into hand gestures and translating them into predefined cursor motion and functions. Techniques like OpenCV – for recognising hand movements and MediaPipe – for processing recognised hand gestures are utilised for capturing and processing gestures for cursor control. With the help of PyAutoGUI, the hand gesture is realised as commands to OS in form of cursor motion. All these processes work in tandem in near spontaneity for lag free experience.

The project utilises a variety of computer vision libraries to achieve Human Computer Interaction (HCI). All these libraries work in real time to map hand landmarks to screen coordinates. Furthermore , with the ability to define Hand Gesture and their utilities to their own specification enables a wide range of customisation. All in all, the system offers a safe and user-friendly contactless navigation experience.

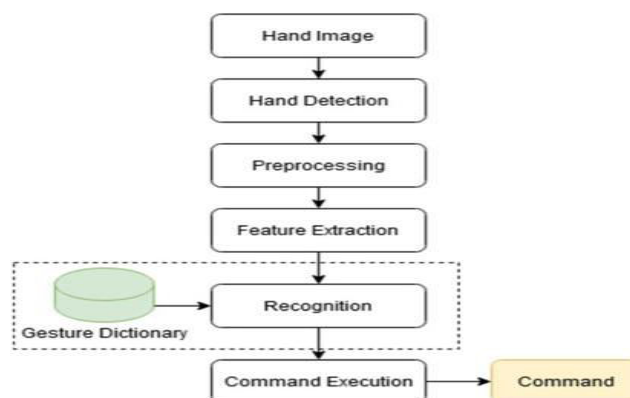


Fig.1. Overall Process of Cursor Control using Hand Gesture

As seen in **Fig. 1**, the hand gesture recognition system follows a systematic procedure over a few steps. First, the hand image is captured. Next, the system detects whether a hand is present in the frame. To improve recognition accuracy, the identified image is then rid of any jitters, refining it in the preprocessing step. Key hand landmarks are identified through feature extraction, and during the recognition phase, they are compared to a predefined gesture directory. This recognition step is vital as it spells for a reliable command execution. The intended control of cursor motion can be observed. The whole system maintains accuracy, reliability and spontaneity by organizing the process in this manner, factoring in various hand variations and environmental conditions such as lighting.

A. HAND DETECTION MODULE

The capability to detect and track hand movements in real-time plays a crucial role in interactive applications for seamless Human Computer Interaction (HCI). Figure 2 represents the key landmarks to be identified of a hand. This can be achieved by using modules OpenCV and MediaPipe for accurate landmark detection, ensuring precise tracking of finger and palm positions.

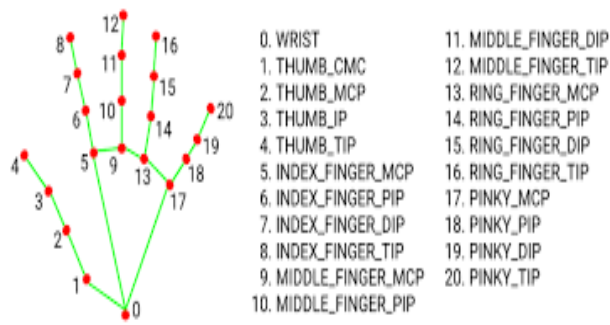


Fig.2. Hand Landmark Mapping

As shown in Fig.2, the system maps the hand landmarks and labels them accordingly which can be later used when defining gestures. This is done by detecting and counting extended fingers, which are identified based on their position relative to PIP joint. An Extended fingers is one whose tip is positioned above the PIP joint. For example, in a flipped image, we compare “x” values for THUMB to determine its orientation.

By creating links between key hand points, the system enables clear and intuitive gesture recognition on a live feed. This is achieved with the help of converted BGR2RGB determining finger states. The model marks essential finger positions like THUMB_TIP, INDEX_TIP, MIDDLE_TIP, RING_TIP, PINKY_TIP. The system adapts to various lighting conditions to guarantee seamless tracking and avoid detection errors. Additionally, it reduces delays, giving it a natural response time to the user. Users can easily interact with digital interfaces and control application with the basic hand gestures.

B. NOISE REDUCTION MODULE

The noise reduction module is essential for achieving smooth and accurate and tracking by filtering out unnecessary movement fluctuations. As shown in Figure 3, techniques like Exponential Moving Average (EMA) help stabilize tracking by softening sudden hand jitters. By applying weighted average in the system ensures a more normal and steady response, making interactions, more fluid and precise.

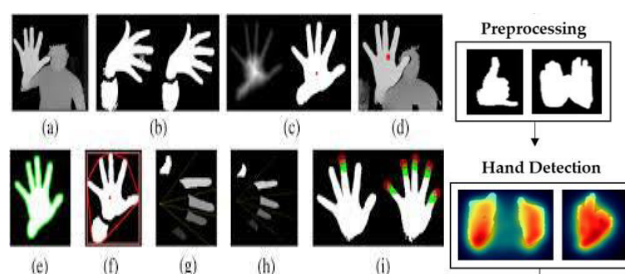


Fig. 3. EMA Noise Reduction for Hand Jitter

As seen in Fig.3, the implementation of formula to smooth out the jitters, filtering out minor, unintentional handshakes that may otherwise cause the cursor to move erratically. Smoothing formula is given by:

$$\begin{aligned}\text{smoothed_x} &= \alpha \times \text{screen_x} + (1-\alpha) \times \text{prev_x} \\ \text{smoothed_y} &= \alpha \times \text{screen_y} + (1-\alpha) \times \text{prev_y}\end{aligned}$$

This approach gives more weight to the most recent movement, while gradually reducing the influence of older data, effectively smoothening out erratic change without introducing noticeable lag.

C. GESTURE RECOGNITION AND ACTION MAPPING MODULE

The gesture recognition module allows the system to understand and respond to specific hand movements, making interactions feel natural and intuitive. As shown in Figures 4,5 detects, predefined gestures, and translate them into comments in real time.

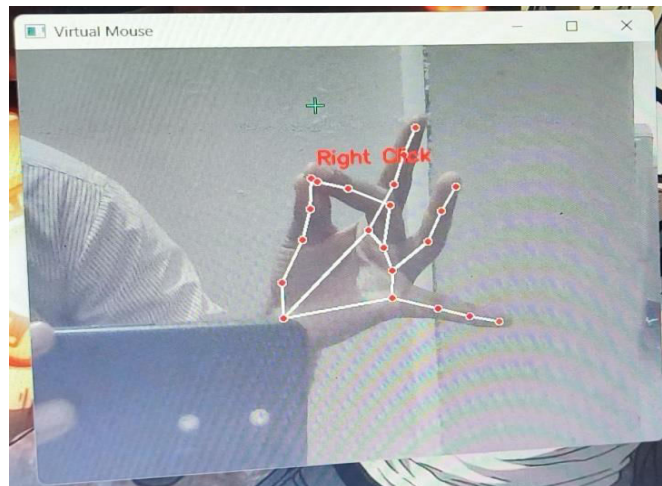


Fig. 4. Right Click Hand Gesture



Fig.5. Left Click Hand Gesture

This module is tantamount to the functioning of the whole system. Based on the identified hand landmarks, it tries to recognize hand gesture. Unless until more than one finger is raised, the system remains idle, as it serves as a countermeasure for accidental gestures. It also includes a cooling mechanism that ensures clicks and scrolls are executed intentionally. The user is privy to what action are they attempting to do, as it displays the recognized action's name on the live feed. Whether clicking scrolling or moving the cursor, the system adapts to certain finger movements, making digital interaction, feel smooth and seamless.

For example, bringing the thumb and middle finger together performs a right-click (**Fig.4**), while touching the thumb and index finger close together triggers a left-click (**Fig.5**). This intuitive interaction method makes navigation effortless, enhancing accessibility and efficiency.

D. CURSOR CONTROL MODULE

This module lets you control the cursor effectively, using simple hand gestures. By tracking the index finger's movement, it smoothly translates gestures into precise, on-screen actions. To keep interaction natural, it uses PyAutoGUI for seamless cursor movement and clicking. This makes navigating a computer intuitive and hands-free.

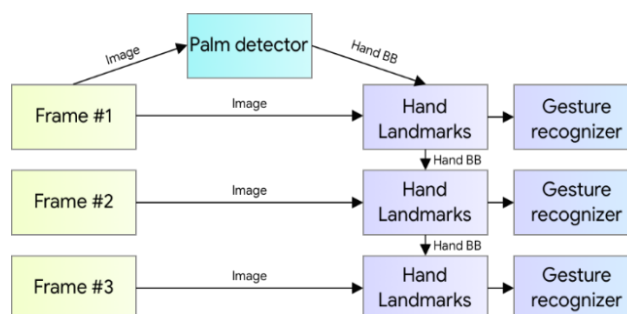


Fig. 6. Gesture Recognizer

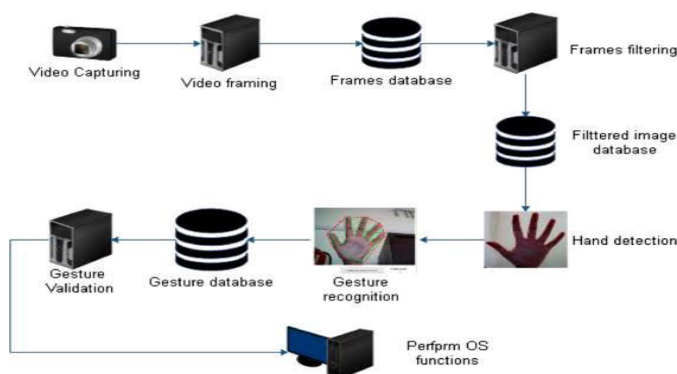


Fig.7. Cursor Control

As shown in **Fig.6** , the process starts with capturing video frames where each frame is analysed to detect and track the hand. A palm detection model identifies the hands position followed by a detailed assessment of individual finger placement. Once the hand landmarks are detected, the system maps them to screen coordinates using NumPy interpolation, ensuring smooth and precise tracking.

As depicted in **Fig.7** , the system then uses PyAutoGUI Ito move the cursor and execute click actions when the fingers come close mimicking a real mouse click. However, quick hand movements can cause the cursor to behave unpredictably. To counter this Exponential Moving Average (EMA) smoothing is applied, reducing sudden jumps and making the movement feel more natural. Beyond just moving the cursor, the system also enables intuitive click and scroll gestures.

IV. SYSTEM ARCHITECTURE

The System architecture provides overview of steps required for translating hand movements into computer commands, enabling users to interact with devices without physical contact. The system ensures accurate and responsive cursor control by real-time tracking and gesture recognition. The flow consists of multiple stages, beginning with hand detection, then pre-processing and feature extraction, finally recognition, which all work in tandem to interpret gestures with high precision. Combined with a predefined gesture dictionary that maps recognized movements to specific commands, ensuring consistency and adaptability across different application.

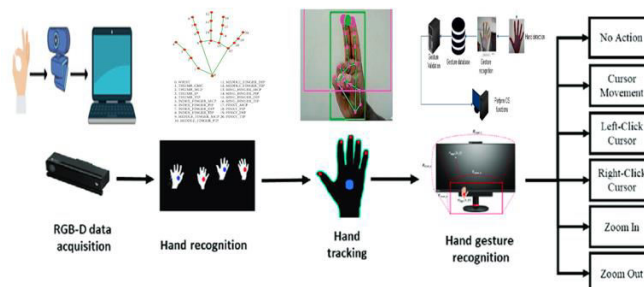


Fig.8. System Architecture for Gesture Control

Gesture based interaction allows for control of digital elements through gestures without need for physical touch. As shown in Figure 8, it explains how hand gestures can be recognised and translated to cursor movements and other actions. This proves as a viable alternative for traditional input methods in the form of mouse or touchpad.

The architecture follows a structured workflow consisting of multiple key stages:

1. **RGB-D Data Capturing and Acquisition**

The process begins with RGB-D camera or WebCam, which captures depth and colour information of the users hand. This step is essential for the algorithm to track the movement of the hands with high accuracy, regardless of the surrounding ambient light conditions. It also mirrors the image to maintain common recognition pattern.

2. **Points Landmark and Hand Recognition**

After the hand is detected, the system then processes the image to locate hand landmarks and key points like fingerprints, joints and palm centre. The hand recognition algorithm, then assigns markers to each of these points, to enable accurate movement tracing. Labelling, this markers helps identify important points necessary to trigger cursor movement.

3. **Hand Tracking based on Hand Movement**

The system attempts to monitor the position and movement of the hand using real-time tracking algorithms. This step is essential in mapping finger positions and maintaining stable tracking even if the hand motion is erratic. Coupled with the application of filtering techniques like EMA, it minimises tracking errors caused by sudden or rapid movements. Any potential jitter is eliminated with the help of stability array.

4. **Hand Gesture Recognition to Cursor Mapping**

The recognised hand gestures are analysed and mapped into predefined actions.

As shown in **Fig.8** , the system recognises gestures and match them to specific commands or functions like :

- **No Action** – The system remains idle when no significant gesture is detected.
- **Cursor Movement** – Moving the hand across the screen moves the cursor accordingly.
- **Left-Click Action** – A predefined gesture such as bringing the Thumb and Index finger together, triggers a Left Click.
- **Reft-Click Action** – Another predefined gesture such as bringing the Thumb and Middle finger together, triggers a Right Click.

V. RESULT AND DISCUSSIONS

The system shows great accuracy in spotting hand signals making it easy for users to work with the interface. This high level of precision helps make gesture control simple and effective, while also cutting down on detection mistakes.

A key feature of this system is how fast it responds. It reacts almost so the cursor moves and acts right away. This quick response is key for tasks that need precision such as virtual navigation new design work, or gaming.

To make motion smooth without any tremor, the system uses Exponential Moving Average (EMA) smoothing. This cuts down on tremor and small hand actions. As a result, the cursor moves smooth giving users better experience and getting rid of any jerky behavior that could harm interactions.

One big plus of this setup is how clean and hands-off it works. Regular input gadgets like keyboards or mice need you to touch them, and they can turn into germ hotspots. But this system cuts out direct contact. That's why it's handy in places where cleanliness is key such as hospitals sterile areas, or public kiosks where it's vital to keep touching to a minimum.

The project is flexible which lets to users make their own gestures for their needs. This flexibility means that it could be adjusted for requirements, with tools to help people with disabilities to smart home control where gestures can start different tasks.

The system can grow beyond its current use. It might be used in gesture-controlled games, AR/VR apps, robot control, and factory systems. The power to control digital screens with gestures creates new ways to interact making tech easier to use and more open to all.

To sum up, the system blends precision quick response smooth movement, cleanliness personal tweaks, and room to grow. This mix creates a good choice over old input devices. As it can spread to new areas, it marks progress in gesture computing. It gives users a natural, hands-off, and flexible way to interact.

VI. CONCLUSION

The gesture-based cursor function control system offers us the advantage of working efficiently by completely replacing the usage of traditional electronic devices that were earlier used by letting us interact with the system, ensuring seamless interaction accurately in real-time. Its potential to recognize the motion and gesture with minimal jitter and adapt to different environments across multiple lighting conditions provides users an edge to use this feature even in low-lit conditions. Furthermore, its contactless and contact-free nature can be integrated into various smart control systems, such as kiosk machines and industrial automation. This can serve as an alternative for the traditional input devices in the healthcare segment, which are often very sensitive for the level of hygiene that should be maintained in these areas. In addition to its current capabilities, the system presents scalability prospects in fields such as gaming, AR/VR, robotics, and even in holographic systems where intuitive controls are much more important. By integrating gesture-based interaction into new technological spaces, it strengthens its value as a forward-thinking solution. Overall, this system has the capability to replace the traditional mouse-based control systems by providing a strong foundation for touchless interaction by offering a smooth, responsive, and adaptable experience.

REFERENCES

- [1] **Ratnesh Kumar, Angad Anand, Prachi Mittal, Shubham, Mahak, Konika Rani (2023)**, GestureCraft: Developing and Evaluating a Virtual Mouse System Using Hand Gestures for Intuitive Human-Computer Interaction, International Journal of Engineering Research & Technology (IJERT).
- [2] **Prithvi J, S Shree Lakshmi, Sohan R Kumar, Suraj Nair, Sunayana S (2023)**, Gesture Controlled Virtual Mouse with Voice Automation, International Journal of Engineering Research & Technology (IJERT).
- [3] **Jagbeer Singh, Yash Goel, Shubhi Jain, Shiva Yadav (2023)**, Virtual Mouse and Assistant: A Technological Revolution of Artificial Intelligence, International Journal of Engineering Research & Technology (IJERT).
- [4] **Wen-Jie Tseng, Samuel Huron, Eric Lecolinet, Jan Gugenheimer (2023)**, FingerMapper: Mapping Finger Motions onto Virtual Arms to Enable Safe Virtual Reality Interaction in Confined Spaces, International Journal of Engineering Research & Technology (IJERT).
- [5] **M. A. Khan, A. Hussain, and R. Kumar (2022)**, Hand Gesture Recognition System Using Computer Vision and Deep Learning, International Journal of Advanced Computer Science and Applications.
- [6] **S. Patel, D. Mehta, and R. Sharma (2021)**, Human-Computer Interaction Using Hand Gesture Recognition: A Review of Existing Systems and Future Prospects, Journal of Human-Centric Computing and Information Sciences.
- [7] **X. Zhang, L. Wang, and H. Chen (2020)**, Biometric Authentication in Gesture-Based Interfaces: Enhancing Security and Usability, IEEE Transactions on Information Forensics and Security.
- [8] **A. Gupta, P. Roy, and N. Das (2023)**, Real-Time Cursor Control Using Hand Gestures: Implementation and Performance Evaluation, International Journal of Computer Vision & Signal Processing.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details