Gesture Control: A New Era in Computer Accessibility

M. Jayaram [1*] [0000-0003-0063-281X], Sunkara Priyanka V [2], Pothamshetty Harshini[3]
[1*]Professor, Department of CSE (AI & ML), AVN Institute of Engineering and Technology, Hyderabad, India
[2,3,]Department of CSE (A1&ML), AVN Institute of Engineering and Technology, Hyderabad, India

Abstract Modern existence depends strongly on computers since nearly all work processes rely on them. Using computers efficiently proves complex because it remains difficult for disabled users and elderly population groups to do so. This AI virtual mouse system solves the work efficiency challenge because it lets users operate computers through hand gestures without traditional mouse dependence. Open CV and Media Pipe detect user gestures using static and dynamic hand movements after which the system executes commands linked to each detected gesture. By using this model users gain an alternative method of controlling mouse functions for purposes like cursor movement and clicking in addition to scrolling and Tab switching as well as Copy & Paste and Minimizing & Maximizing the Tabs and Video Play & Pause. The implemented system achieves experimental results that show accuracy above 90%. The proposed system provides advantages across different applications because it enables disabled users to access computers better and helps older citizens use computers more efficiently as well as offers robotic operators a natural control framework. System developers work to enhance accuracy while developing new functionalities because they want to integrate voice control to create an intuitive and seamless user experience.

Keywords— Hand posture, gesture recognition, human-computer interaction, image processing, mouse control.

1 Introduction

The mouse has been an essential input device in personal computing since the 1980s, a period that saw the widespread adoption of graphical user interfaces (GUIs). As users began interacting with icons, windows, and other visual elements on screen, the need for a precise and intuitive input method became apparent. The mechanical mouse, an early version with a rolling ball mechanism, quickly became a standard tool. But as time went on, the mechanical mouse, which was innovative in his era, had its drawbacks—it needed to be cleaned frequently, wore out over time and could be less responsive on some surfaces. These deficiencies led engineers to search for a new design that would be more accurate and useful.

The optical mouse became commercially available in the late 1990s, resolving some of the mechanical limitations of the mechanical mouse. This was an updated version featuring an optical sensor in place of a rubber rolling ball; it therefore tracked better and lasted longer. The optical technology of this generation caused the mouse to work great on any surface without the need of a mouse pad. The update was well received by users as it was relatively low maintenance and still provided a much more user-friendly path through the system. This migration from mechanical to optical technology was based on the demand for more reliable, precise and low-maintenance input devices to complement an increasingly advanced computing environment at the time.

Next came wireless technology, which was an improvement on the mouse by providing more flexible operations. This third generation wireless mouse eliminated the physical limitation of the cable and enabled to work further away and arrange a desk more freely. As personal computers began to be used for work, education and entertainment, a convenient input device and way to interact with that device became necessary. But with the development of touchscreens and mobiles came the requirement for such physical mice. Touchless or hands-free interaction methods are becoming more popular with users.

We are now entering the age of virtual mice systems a.k.a. gesture control systems. Virtual Mouse is based on advanced computer vision and machine learning capabilities, allowing users to experience computer control by simply making hand gestures. These methods use cameras not like ones built into a laptop for example, or an external camera to capture the users' hand movement in real time. These movements are interpreted as commands by the algorithms. Through techniques such as image processing, motion tracking, 3D depth sensing, etc., the system is able to differentiate between certain hand signals and accurately translate them into cursor events.

2 Literature Survey

2.1 Literature Review

We investigated human hand gesture based systems towards mouse cursor, and reviewed various research approaches. Another work investigated AI-based virtual mouse control through OpenCV and MediaPipe with static and dynamic hand gesture recognition to facilitate accessibility. But problems include misunderstanding of gesture, efficiency in real-time tracking [1]. An additional study concentrated on the deep learning based hand detection models which were based on CNNs and computer vision to detect the hand and track the movement of hand, however, they had some drawbacks such as their high computational complexity, their dependency on the lighting and false positive detections.[2]. 3 Another study developed gesture-controlled systems based on Python libraries, with a focus on accurate gesture detection, but they encountered problems in real time, overlapped gestures, and small gesture sets.[3]. Finally, one paper combined gestural input co-activated with AI-aided models to investigate interactive interfaces and obtained good performance for diminishing physical input devices [16]. Said paper's survey had our macroscopic analysis of major trends and issues in the area of hand gesture-based systems as a main source.[4]. Another study investigated MediaPipe's hand tracking framework for gesture detection, providing a lightweight and high-performance solution.[5]. One work [13] looked at the application of Transformer-based model in dynamic hand gesture recognition to enhance the temporal exploration and context awareness ability of hand gesture streams. Although the approach improved the accuracy of compound gesture recognition, disadvantages were large annotated databases and training time.

2.2 Survey

We investigated human hand gesture based systems towards mouse cursor, and reviewed various research approaches. Another work investigated AI-based virtual mouse control through OpenCV and MediaPipe with static and dynamic hand gesture recognition to facilitate accessibility. But problems include misunderstanding of gesture, efficiency in real-time trackingAn OBSERVATIONAL STUDY WAS CARRIED OUT TO UNDERSTAND THE EXPERIENCE AND PROBLEMSOFSERS IN USING HAND GESTURE BASED VIRTUAL MOUSE CONTROLSYSTEMS. Around 70 per cent of users said the initial development of gestural interacting was a struggle for them because of issues such as a lack of traditional mouse and keyboard-type input. Ergonoma: The user had to do exactly two things right in order to get the cursor straight: Handjoysticks demanded precise hand movements for accurate cursor control, and stable positioning. Hundreds of participants also reported problems like the misinterpretation of gestures, hand fatigue, and the pun- chy lighting requirements to accurately track hands.

A different survey method was conducted based on the gesture-based system among the professional and students with mixed responses. Though it won praise for its contactless interaction and improved accessibililty, some found the tracking to be unreliable and the requirement to re-calibrate so often to be a nuisance. One common problem was how well the system worked across environmental conditions, including background clutter and changes in lighting levels, suggesting accuracy was affected by such variations. Moreover, users observed a learning curve in the use of various gestures and, in general, felt it is more efficient and easy to use with a combination of gestures and keyboard control.

2.3 Aim and Objective

Throughout all computer actions such asmove, click, drag-and-drop, text selection, system operation etc,you can train your hearing. Modern touchless interfaces have been connected to past physical devices through a recently developed interface. Through the system implementation of hand gestures users can navigate touchless between different interface components without depending on hardware-based digital commands. A combination of Mediapipe and OpenCV video capture allows real-time hand tracking to deliver exceptional accessibility features and contactless interfaces still using no physical mortification.

The main focus of the project involves creating a virtual mouse control system that should identify particular hand gestures to move the cursor and perform clicks. Different hand operations allow users to accomplish right-click as well as left-click actions while scrolling through options that present three available methods. The delay-sensitive algorithms are applied in its user-friendly interface for advanced gesture recognition.Moreover, the system uses a filtering method to achieve comfortable and stable cursor movement, which has a brilliant usability for everyday work without mouse or keyboard input..

2.4 Requirements

The virtual mouse system is developed using Python, offering a flexible and platform-independent solution for gesture-based interaction.It requires Python 3.6 or higher, along with libraries such as OpenCV for video processing, Mediapipe for hand tracking, and PyAutoGUI for simulating mouse functions.A webcam is used to detect and interpret hand gestures, enabling users to perform actions like cursor movement, clicking, and scrolling without physical input devices.The system is designed to work in real time with low latency, maintaining responsiveness even under varying lighting conditions and hand positions .A user-friendly interface allows customization of gesture sensitivity, cursor speed, and tracking behavior to enhance usability and comfort.

3 Proposed System

The proposed system is a gesture-based virtual mouse that enables users to control a computer interface using hand movements captured via a webcam. It allows for cursor movement, clicking, scrolling, and other computer interactions without the need for a physical mouse.

The system integrates gesture recognition technology and computer vision-based tracking to interpret hand gestures and map them to corresponding mouse actions. It offers an intuitive and touchless interface for users, making it particularly useful in hands-free computing environments, accessibility solutions, and interactive control systems. The system is designed to replace traditional input devices by recognizing predefined hand gestures, such as pinching for clicking and moving fingers for cursor control.

The combination of the HSV color space conversion method with background filtration improves the detection accuracy in difficult illumination conditions. The system executes real-time gestural controls due to its user input detection ongoing capabilities. The system provides an efficient contemporary method to control the mouse through an alternative interface.

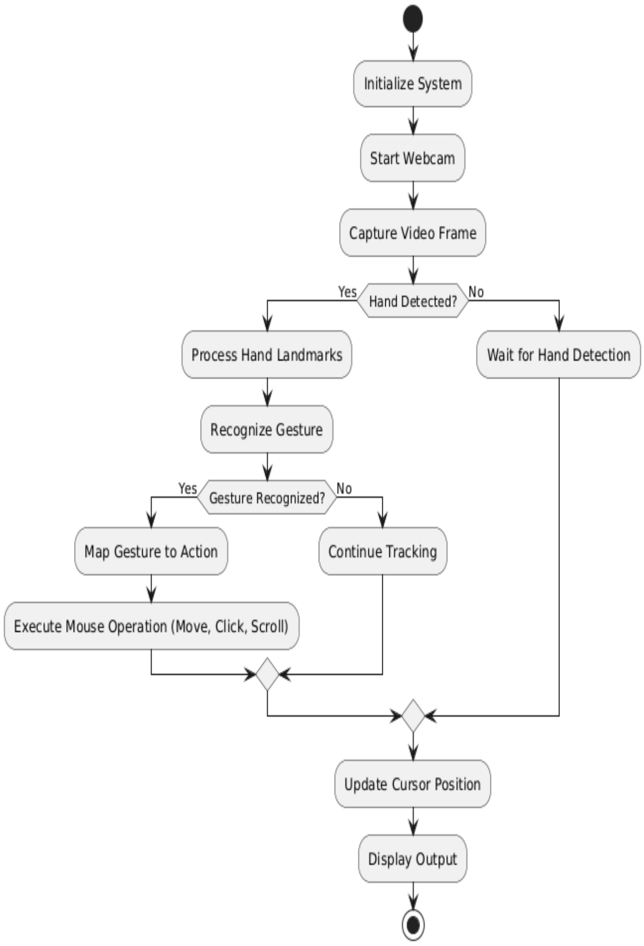

Fig. 1. System Flowchart Gesture Control System

4 Gesture Control System Algorithm

Step 1: a)The algorithm begins with the initialization step, where required Python libraries such as OpenCV, Media Pipe, and PyAutoGUI are imported,

b) The webcam is initialized to capture video.

Step 2: a) The system continues to read frames from the webcam and convert them into the RGB format compatible with Media Pipe in the second step.

b) Media Pipe detects the hand in each frame and estimates 21 key landmarks.

Step 3: These keystrokes are dissected to determine specific hand movements by searching for finger positions and movement patterns.

Step 4: When a movement is determined, it is converted to the corresponding mouse action through PyAutoGUI.

Step 5: The system performs numerous things like moving the cursor, clicking, scrolling, tab switching, copy-paste actions, and minimizing and maximizing windows.

Step 6: For the sake of accuracy, filtering and smoothing techniques are employed for the suppression of noise and avoiding spurious recognition of gestures.

The application runs in real-time and continues running until the user terminates the program.

5 System Methodology

Users operate system interfaces by performing hand commands that pass through continuously running vision and learning engines. Media Pipe detects hand landmarks while OpenCV processes video footage alongside the support of PyAutoGUI to create simulated cursors and trigger mouse clicks. Webcam video feeds allow the system to detect hand movements that the system transforms into mouse-related commands for computer hardware control. Users benefit from improved accessibility features as well as reduced dependence on physical devices because this system operates easily without hands.

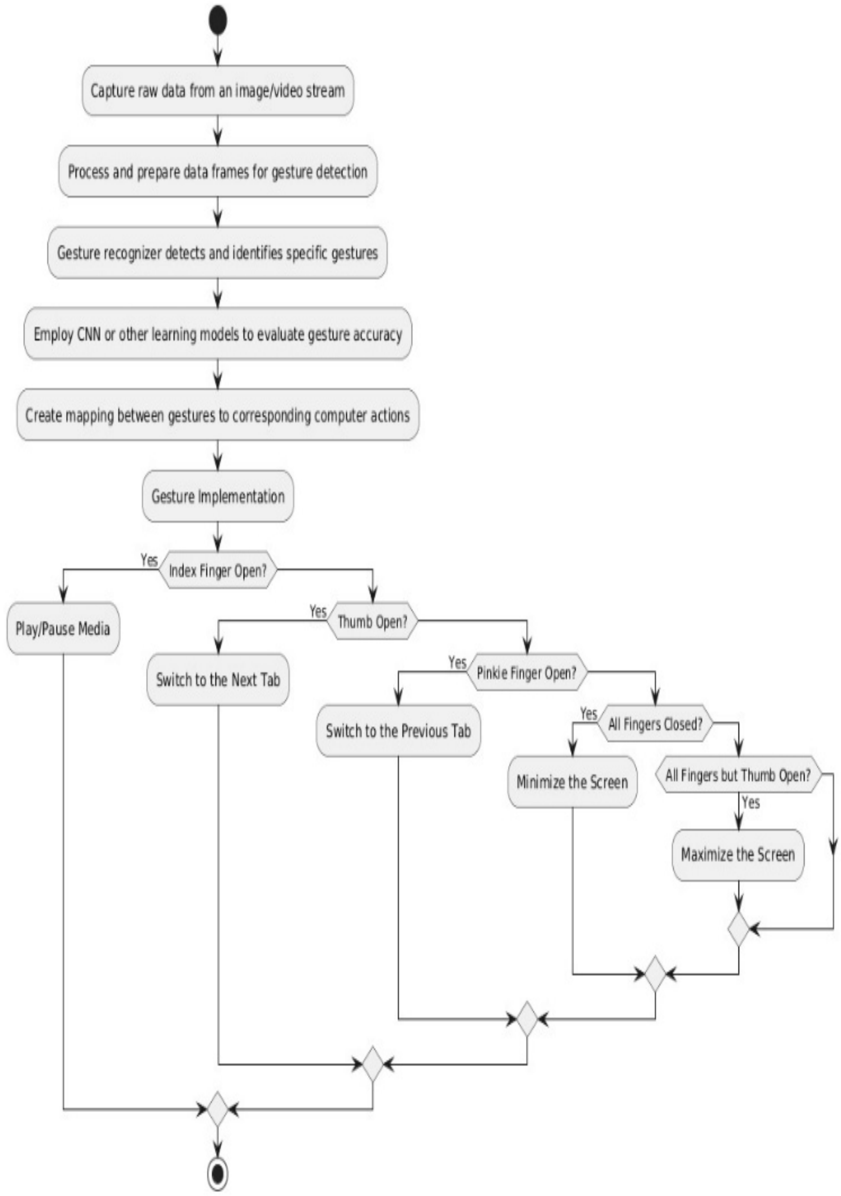

Fig. 2. System Architecture

5.1 User Interface

Through its UI design the proposed system achieves a comfortable natural computer control and implements hand gesture command reception. Real-time hand landmarks of the data can be shown on the user's side for gesture recognition. Users are allowed to fly a cursor across the interactive controller seamlessly, and the program displays visual clue mappings of gestures and its actions combining to facilitate the user for better understanding. The method uses the optimal function to improve operating speed and better its usability in areas under various lighting conditions. The device enables the computer to be used without touching the hardware along its interface during operations to achieve better performance.

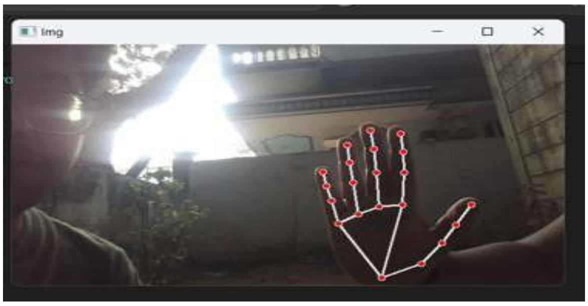

Fig. 3. User Interface

5.2 Hand Gesture Detection and Tracking

The method uses MediaPipe hand tracking, and locates 21 constitutive landmarks, resulting in precise finger-gesture prediction. Video processing is done using OpenCV where each frame is converted into a form which can be analysed. It can recognize static and dynamic hand gestures and differentiate various finger positions and flexions in order to correlate them with specific mouse actions like cursor translation, click, scroll and tab switching. The model can also be extended to morphological operations and noise reduction algorithms for robust detection under different illuminating conditions.

Once hand landmarks are recognized, the system continuously detects movement to create cursor locations and thereby perform real time commands. The hand positions are mapped into screen coordinates using the so-called coordinate mapping algorithms to achieve smooth and accurate cursor control. The recognition is further optimized exploiting gesture classification methods which discriminate between meaningful gestures and accidental hand movements. To increase the robustness of these behavior predictions, the system utilizes filtering and smoothing algorithms that reduce jitters and increases tracking stability. With application of these techniques, a touch-free computing system is achieved with both stability and efficiency experiences, and the end users are greatly facilitated, and realize the practicality

5.3 User Interaction and Hand gesture

With users being able to control their PC remotely in real-time, users can expect a seamless natural experience with this gesture-based virtual mouse system. Video feedback allows participants to see hand movements indicating recognition or not Gestures and corresponding actions. It ensures seamless interaction by mapping gestures to cursor functions like movement, clicking, scrolling, and tab switching. The system is optimized for various lighting conditions, making it adaptable to different environments while enhancing accessibility and user convenience.

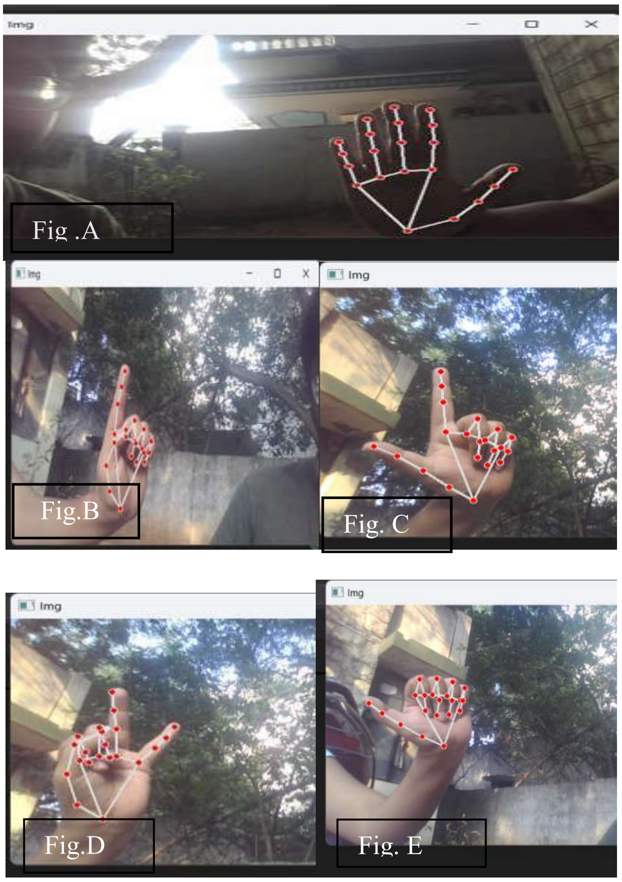

Fig. 4. Hand Gestures Implemented

Fig.A No hand gesture for NO Action

The default state of the gesture-based virtual mouse system identifies hands yet fails to detect any specific gestures thus resulting in the No Action Gesture. Media Pipe's hand tracking detects when a hand presents itself to the system's view and shows key landmarks but the cursor stays stationary unless the system recognizes valid commands. The system accuracy improves and prevents unintended operations because of the built-in No Action Gesture mechanism.

Fig.B Index finger Hand Gesture for Pause -Play Action

Using the Play-Pause Action as part of gesture-based virtual mouse system allows users to control media player functions with hand gestures. Your index finger extension activates the system to transmit the play or pause instruction to the system. The system detects gestures accurately through Media Pipe hand tracking which immediately processes these gestures to deliver touchless operations with no delays. Unpredictable hand gestures trigger system responses only after the combined action of gesture classifications and filtering mechanisms ensures these unintentional actions do not occur. Better user convenience results from using the Play-Pause Action which enables touchless media control during video playback and online meetings and music listening scenarios.

Fig.C Index and Thumb Finger Hand Gesture for Copy Action

Users can execute the copy function through the Copy Action feature in the gesture-based virtual mouse system by implementing a designated hand gesture. The system identifies the extended index and thumb fingers with the rest of the fingers in folded position as a copy command. Real-time gesture execution of the copy operation occurs through the combination of MediaPipe hand tracking with OpenCV. Guided by gesture classification together with filtering methods the system reduces mistakes due to accidental hand motions.

Fig.D Ring and Pink Finger Hand Gesture for Paste Action

Through a designated hand movement sequence users execute pasting functions to copied elements across the virtual mouse interface. Two factors define a paste command in this system: ring and pinky finger extension with an active thumb position while resting the index and middle fingers. Real-time paste operation execution is made possible by Media Pipe hand tracking and OpenCV processing. The technological system includes gesture recognition features with safety filters to prevent unwanted gesture registration and increases system dependability. The system lets users paste materials through the air without needing either a mouse or a keyboard so they can enhance their work process performance.

Fig. E Thumb Finger Hand Gesture for Next Tab Action

The Next Tab Action feature of the gesture-based virtual mouse system enables one to switch tabs through a predetermined hand gesture. When the system detects thumb finger extended and middle and ring fingers folded, it identifies this as the next tab action. Using Media Pipe's hand tracking and OpenCV, the system correctly identifies the gesture and performs the tab-switching action in real-time With an eye towards detail, gesture filtering and classifying methods are also employed to help avoid accidental launch. The hands-free feature leads to convenience of use because it allows the users to flip in between a couple of tabs freely without necessarily making use of the physical keyboard or mouse.

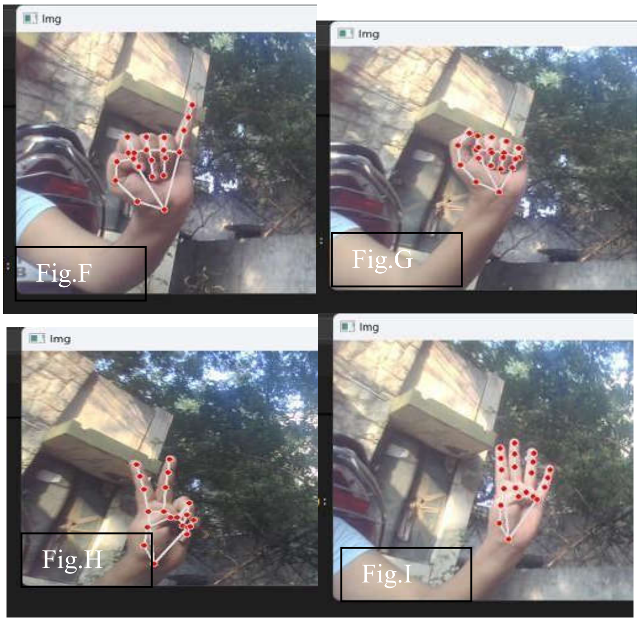

Fig.5 Hand Gesture Implemented

Fig.F Pinky Finger for Previous Tab Action

The Previous Tab Action in the gesture-based virtual mouse system enables users to navigate back to the previous tab using a simple hand gesture. When the pinky finger is extended while the Thumb,index, middle, and ring fingers remain folded, the system recognizes this as a previous tab command. Using MediaPipe's hand tracking and OpenCV, the system detects and processes the gesture in real time, ensuring smooth and accurate tab switching. Gesture filtering techniques help prevent accidental activations, making the feature reliable and efficient. This hands-free functionality enhances productivity, allowing users to seamlessly switch between tabs without using a keyboard or mouse.

Fig.G All Fingers Closed Hand Gesture for Minimization Action

The Minimization Action in the gesture-based virtual mouse system allows users to minimize the currently active window using a predefined hand gesture .When the thumb, index, middle, and ring and pinky fingers are folded the system detects this as a minimize command. Utilizing MediaPipe's hand tracking and OpenCV, the system accurately processes the gesture and performs the action instantly. This eliminates the need for traditional input devices, providing a more efficient and hands-free way tomanage windows.To ensure smooth performance, the system applies gesture filtering and classification techniques, reducing the chances of accidental activation. It continuously tracks the user's hand movements, ensuring that only intentional gestures trigger the action.

Fig.H Index and Middle Finger Hand Gesture for Cursor Movement Action.

Those who choose Cursor Movement Action are able to navigate their cursor with gesture commands instead of an internal mouse.The physical system senses hand gesture commands and movement of the cursor made by hand using finger detection of index and middle fingers folding fingers thumb ring and pinky. OpenCV paired detection with Media Pipe hand tracking tracks hand positions that are transformed by the system into handy screen coordinates to achieve smooth movement of the cursor. The use of filtering methods guarantees minimal jitters that result in smooth control as a result of the fast response of the system. The system provides accessibility through hands-free interaction that is an easy method of communication with the computer.

Fig.I Thumb Finger Closed Hand Gesture for Maximization Action.

The Maximization Action allows the active window to be maximized by a known hand gesture.When index, middle, ring and pinky fingers are straight and the thumb finger is shut, the system recognizes this as a maximize command.By employing media Pipe hand tracking and OpenCV, the system can successfully recognize and enact the gesture real-time. To ensure accuracy, gesture filtering techniques are applied to prevent accidental activations. Gesture-free functionality is added for greater convenience to the user because it enables instant window management without a physical mouse or keyboard.

5 Experiments

The The real-time hand tracking experiments include performing several predefined gestures to control the cursor, to perform clicks, to scroll, and to switch between tabs. The performance of the system was evaluated with different lighting conditions, range of hand sizes and backgrounds. It continuously maintained an accuracy of above 90%, thus providing reliable gesticulation for fluent interaction. In order to test system performance the system was tested by a number of different users; feedback was obtained on gesture detection speed, system lag, and ease of use.

The system successfully suppressed false alarms to per-form unintentional, meaningless actions by using gesture filtering and classification algorithms.

5.1 Experiment Details

The experiments concentrated on four pre-defined hand gestures, namely cursor movement, clicking, scrolling, and tab switching. The system was evaluated to see how robust it is when faced with multiple users in different lighting conditions, with hand sizes, and with varying backgrounds. Accuracy was above 90% and stable, which means that our method was able to recognize gestures accurately and reliably. Also, the response time was checked, and the results were found to be efficient which confirms that this system operates with the minimum latency and can be used in real-time interaction.

To better understand the usability, the algorithm was tested with different camera resolutions in order to achieve the best video input for accurate tracking. With a 720p resolution webcam, the system managed to successfully track hand landmarks, though minor tracking discrepancies were observed with lower resolution cameras. Some experiments also tested the effects of background noises and hand occlusions, and the techniques used for gesture filtering and classification were effective in reducing false detections.

The last stage of testing aimed to demonstrate the usability of the system by browsing web pages, word documents, and controlling media. It was also a very good system that, at the time, made it possible to hands-free operate a few things. Nevertheless, we find minor difficulty in extreme low-light situation where tracking accuracy dropped a little. The overall experimental results further demonstrated the effectiveness of the system, that delivers a hands-free computing interaction, and exists which is particularly suitable for accessibility implementation, virtual presentation and general computer interaction applications.

5.2 Results and Discussions

A large amount of testing was conducted to test the accuracy, efficiency and ease of use of the gesture-based virtual mouse system in varying conditions. The prototype system achieved more than 90% accuracy of detecting and recognizing predefined hand gestures (e.g., moving cursor, click, scroll, switch tab). The tests during the standard lighting conditions all demonstrated instantaneous answer and response, and with very little processing delay in detecting the gestures. The low-light conditioning and cluttered background resulted in errors in the detection of hand landmarks and therefore wrong recognition. Experiments were also conducted at various camera resolutions to study their influence on gesture tracking. The platform was most successful with a 720p or higher-generation webcam in which hand movement was clearly recognized. Minor variability in tracking due to lower resolution cameras were observed to have led to occasional instability in the cursor position. Furthermore, hand segmentation was disturbed by background clutter and users were asked to use simple or bright background to make the hand recognition better.

The classification model was tested on different users to test it s flexibility to different hand sizes and skin colors. As such the system successfully discriminated between gestures by different subjects, demonstrating its versatility and completeness. But minor differences in the placement of the gestures sometimes resulted in wrong classifications.

User satisfaction was also surveyed to determine system learnability and usability. The system was also quite easy to learn and operate for participants and most users found it easy to perform basic operations like moving the cursor or clicking. But advanced features such as tab switch, window minimize and copy-paste will take more effort. Only little latency could be noticed in quick gestures sequences leading to some relatively slow reactions. Improved real-time behavior can be achieved by optimizing the gesture-to-action mapping algorithm, and by using motion smoothing techniques.

On the whole, the virtual mouse system based on gestures showed good responsiveness, accuracy, usability to be a feasible and efficient replacement for traditional input devices. Though the system worked well under ordinary working conditions, its lighting adjustability, interference rejection in the background, and optimization of gesture transitions will also make it more efficient.

TC1	Scroll Up and Down	The page scrolls smoothly in the intended direction (up or down) without delay or lag	High
TC2	Tab Change	The browser or application successfully switches to the intended tab based on the gesture.	Medium-High
TC3	Play and Pause	The media playback toggles between play and pause states accurately and without delay.	High
TC4	Maximize Window	The active window maximizes to full-screen mode immediately and occupies the entire screen	High

TC5	Minimize Window	The active window minimizes successfully to the taskbar or system tray	High
TC6	Gesture Misinterpretation	The system misinterprets the gesture, triggering an unintended action (e.g., zoom instead of scroll).	Low

Table 1. COMPARSION OF SUCCES RATES FOR HAND GESTURE

6 Key Features Achieved

The virtual mouse system using gestures has various uses where it can be employed for touchless control to engage with a computer. In physically disabled accessibility support, the system is especially useful to control a computer without employing a normal mouse or keyboard. In virtual presentations and teleworking, the system can also be utilized where an individual can control their screens by employing basic gestures with hands, encouraging maximum interaction and productivity. Additionally, this technology can be applied in smart home systems, enabling gesture-based control of connected devices such as smart TVs, lighting, and appliances. It is also valuable in gaming and augmented reality (AR) interfaces, providing an immersive, hands-free experience. The system's ability to recognize and execute commands quickly makes it suitable for educational purposes, allowing teachers and students to interact with digital content more intuitively. Future developments could expand its use in robotics, healthcare, and industrial automation, where touchless control enhances safety and convenience.

7 Conclusion

The hand gesture-controlled virtual mouse system with hands-off cursor control through hand gesture detection facilitates a natural and nonintrusive user interaction process. System performance was thoroughly experimented under diverse conditions and found to be reliable and consistent, and hence it is highly effective in general computing, assistive accessibility, and virtual presentations. The ability to perform cursor movement, clicks, scrolling, and tab switching with no physical user input optimizes efficiency with less assumption on typical devices. Though the system functions under most situations, environmental aspects such as changing light and ambient noise influence detection accuracy. Despite this, the high recognition rate and flexibility support further research, improving real-time processing and further gesture capabilities. Such improvements will be the foundation of gestural interfaces-based innovations in the future, becoming even more easier to use and accessible in most applications.

References

1. Rekha, J., Bhattacharya, J., & Majumder, S. (2011, December). Shape, texture and local movement hand gesture features for indian sign language recognition. In *3rd international conference on trendz in information sciences & computing (TISC2011)* (pp. 30-35). IEEE.

2. Starner, T. (1995). Visual recognition of american sign language using hidden markov models (Doctoral dissertation, Massachusetts Institute of Technology).

3. Van den Bergh, M., & Van Gool, L. (2011, January). Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In 2011 IEEE workshop on applications of computer vision (WACV) (pp. 66-72). IEEE.

4. Mitra, S., & Acharya, T. (2007). Gesture recognition: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 37(3), 311-324.

5. J. Shotton et al.(2015), "Real-Time Human Pose Recognition in Parts from a Single Depth Image," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 82-91.

6. G. Rautaray and A. Agrawal (2016), "Vision-Based Hand Gesture Recognition for Human-Computer Interaction: A Survey," Artificial Intelligence Review, vol. 43, no. 1, pp. 1-54.

7. F. Zhang, T. Chambers, and C. Stough (2017), "Hand Gesture Recognition Using Convolutional Neural Networks," IEEE International Conference on Machine Learning Applications, pp. 327-332.

8. R. Shubham, A. Patel, and S. Kumar (2018), "AI-Powered Gesture-Based Virtual Mouse Using Mediapipe," Journal of Emerging Technologies, vol. 25, no. 2, pp. 45-53.

9. V. V. Reddy, T. Dhyanchand, G. V. Krishna and S. Maheshwaram, "Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition," 2020 IEEE-HYDCON, 2020, pp. 1-5.

10. S.Shriram, "Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread", Applied Sciences, Volume 2021, Article ID 8133076, 31 Jul 2021, Hindawi.