International Journal of Emerging Research in Science, Engineering, and Management
Vol. 2, Issue 1, pp.288-297, January 2026.
www.ijersem.com eISSN – 3107-9075

AI Virtual Mouse Using Hand Gestures

[1]B.Hima Bindu, [2]Godugu Thejaswini, [3]Kovuru Ramalakshmi, [4]Konga Uma Maheswari, [5]Sompalli Pavithra, [6]Nemalla Venkateswara

Department of CSE, Siddartha Institute of Science and Technology, Puttur, India.

[1]himabindubukkapatnam@gmail.com, [2]Goduguthejaswini@gmail.com, [3]kovururamalakshmi@gmail.com, [4]Umakonga12@gmail.com, [5]pavithrasompalli@gmail.com, [6]Venkynemalla2005@gmail.com

Abstract: This represents an increasing trend towards natural and touchless interfaces, driven by leaps in computer vision and AI. The mouse device itself imposes serious limitations, above all in sterile environments, accessibility-focused scenarios, and situations that absolutely require hands-free control. Deep learning-based gesture recognition overcomes this through virtual mouse systems-interpreting hand movements captured via a webcam into cursor actions. With an AI-enabled Virtual Mouse System, intuitive, touchless movements, clicks, drag, and scroll operations are provided through simple hand gestures. It enhances the sense of accessibility, hygiene, and allows for a seamless experience that proves quite valuable in healthcare, gaming, AR/VR applications, and other smart and interactive settings.

Keywords: Hand Gesture Recognition, AI Virtual Mouse, Computer Vision, Touchless Human–Computer Interaction, MediaPipe Hands.

1 INTRODUCTION

Human-computer interaction has gradually moved from traditional mechanical input devices to more natural and intuitive ways of intelligent interaction. While the mouse and keyboard have been the leading tools for inputting information into computers, they are at a disadvantage in many situations where physical contact is either inconvenient, restricted, or not hygienic. The integration of digital systems into healthcare, automation, virtual reality, and smart environments has placed touchless, gesture-based interfaces in the limelight [1]. Thanks to recent advances in the areas of artificial intelligence, computer vision, and deep learning, it is possible to endow virtual tools with the capability to interpret human gestures with a high degree of precision, thus opening possibilities for developing a completely new generation of intuitive, touch-free interfaces.

The idea of an AI-driven virtual mouse eliminates the use of a physical mouse by capturing hand motions through a standard webcam [2]. The user will control the cursor in mid-air with natural hand motions, without mechanical movement over a flat surface. In this respect, it forms a highly flexible interaction medium that can operate in multiple contexts without additional hardware. Systems like these use deep learning-based gesture recognition, landmark detection, and motion tracking to interpret real-time video frames and translate them into interactive commands like moving the cursor, clicking, dragging, and scrolling. MediaPipe Hands, OpenCV, and Convolutional Neural Networks have been the technological enablers for the exact detection of fingertips, posture of the hand, and trajectories of movements. Such a touchless virtual mouse system has considerable advantages in every environment where hygiene is utmost necessary. To give examples, hospitals, laboratories, and cleanrooms require at least an amount of surface contact so that the sterile conditions of their operations are retained.

A gesture-controlled virtual mouse avoids contamination since the user is not physically interacting with any device. In other accessibility situations, this allows personalization of gesture controls according to individual motor disabilities or limits on hand mobility. They will thus be able to use computers more comfortably and without having to rely on others for support, hence offering better digital accessibility and therefore inclusion. At the same time, while immersive technologies such as AR, VR, MR, and smart homes are under rapid development, gesture-based interfaces are becoming highly integrated as a natural interaction method with virtual objects. Unlike controllers, gesture recognition allows users to manipulate UI elements through simple and natural motions, thus offering greater immersiveness and user experience. The AI Virtual Mouse System enacts these principles in that, effectively, intuitive human-machine communication is allowed through real-time computer vision without special hardware such as data gloves, infrared sensors, or depth cameras. It requires only a normal RGB webcam, which greatly reduces the cost and increases accessibility [2][3].

Another motivation for developing an AI Virtual Mouse comes from the fact that deep learning has reached such a mature level that gestures could be robustly recognized under extreme variability of lighting conditions, complex backgrounds, and significant variations in the shape of hands. It was exactly this kind of noise and variation that frustrated traditional computer vision technologies, while modern neural-network-based detection models proved to be reliably accurate. Deep-learning methods are trained on enormous datasets containing various hand poses and gestures; thus, the virtual mouse will function accurately, whatever the size of your hands may be, whatever your skin color is, and however it wave them. Additionally, machine learning allows a scope for customization and adaptability, hence enabling the system to learn new gestures and personalize control for different users. In turn, this develops the need for hands-free interaction tools due to remote work and online learning. Anyone interested in virtual presentations, coding sessions, or online classes, combined with practical tasks, would definitely opt for touchless controls since it is easy and more efficient. Effortless navigation, unrestricted by physical limitations, with the AI Virtual Mouse smoothes digital workflows, making them flexible; it can also be integrated with voice commands for enhancing multimodal interaction and increasing versatility in the system.

The two major objectives of this paper are designing a practical, real-time virtual mouse that can perform basic cursor operations based on simple hand gestures. Major functionalities to be dealt with are hand detection, identification of its postures, determination of its gesture pattern, mapping of gestures into mouse events, and keeping the cursor's motion smooth. Minimum latency with maximum accuracy is assured by the system in order to let users work with an intuitive experience. Other objectives are adaptability in different environmental contexts, keeping computational complexities at lower levels, and providing a user-friendly interface for calibration and testing of the system [4]. The importance of such a system goes beyond mere convenience: gesture-based AI interfaces represent the core of how the future of computing will bridge the gap between human and machine. As smart devices continue to grow, gesture control is intrinsic to the interactive ecosystem that underpins an expanding array of applications in automation, robotics, gaming, rehabilitation, and assistive technologies. The AI Virtual Mouse paper contributes within that frame of technological evolution to demonstrate a practical, low-cost AI-driven solution that can easily replace traditional input devices.

Conclusion: AI Virtual Mouse Using Hand Gestures opened a new frontier to modern, hygienic, and inclusive ways of interaction with computers. On this platform, computer vision-fortified with deep learning-shall be able to recognize in real time natural human hand gestures and set up seamlessly the path of communication between users and devices. Presupposed areas of applications will include quite a wide variety of fields, while development promises to mark the rise in interest of mankind in the evolution toward increasingly intuitive and intelligent digital interfaces.

2 LITERATURE SURVEY

Computer vision, AI, and gesture recognition have totally changed the face of human–computer interaction. The research on gesture-based interaction in the early times depended on hardware-dependent solutions, such as sensor gloves, accelerometers, infrared devices, and depth-sensing cameras. The results of systems like the Nintendo Wii Remote, Leap Motion Controller, and Microsoft Kinect showed very good promise for gesture-based interfaces but utilized specialized hardware, thereby limiting their scalability and accessibility. While these devices have, in principle, allowed for high accuracy and low latency, they lacked affordability and convenience related to ordinary webcams. Very soon thereafter, research shifted toward camera-based gesture recognition in an effort to reduce hardware dependency and increase accessibility for the general user. Computer-vision-driven gesture control rapidly gained relevance, especially with developments in the domain of deep learning that drastically improved the precision of hand and motion detection [5].

Traditional computer vision methods for gesture recognition involved edge detection, color segmentation, background subtraction, and contour extraction. Early works tried to track hand position either by detecting skin-colored blobs or by considering the movement patterns using optical flow. While these were at the foundation, they were highly susceptible to environmental factors concerning variation in lighting, background clutter, and skin-tone diversity. Consequently, such systems were not robust and could not handle complex gestures or real-time interactions reliably. Researchers realized the limitation of handcrafted feature extraction and began to explore learning-based methods that would generalize across diverse conditions. That was the beginning of modern gesture recognition research.

Deep learning completely revolutionized gesture recognition by automatically extracting features and performing robust classification [5][6]. The most significant contributions in this respect have come from Convolutional Neural Networks, which have shown unparalleled accuracy in detecting objects, estimating hand pose, and classifying gestures. Later, architectures such as YOLO, Faster R-CNN, and SSD helped to detect the presence of hands within a live video stream more efficiently and accurately. However, deep landmark detection frameworks brought even more vital changes. For instance, Google's MediaPipe Hands model offers a lightweight yet extremely accurate solution that can detect the 21 landmarks of a hand in real time using a standard webcam. So far, it has gained prominence because it enhances the reliability of virtual mouse systems whereby tracking the exact position of fingers, joints, and the palm is done with high accuracy. This kind of model improves the classification of gestures, too, by providing comprehensive pose information rather than just depending on contours or skin segmentation. Most of the research in this area is oriented to either static gesture classification or dynamic gesture recognition. Examples of static gestures include an open palm, closed fist, and pointing finger.

Dynamic gestures are sequences of motion. Virtual mouse systems rely significantly more on dynamic gestures because the nature of the task is continuous cursor control. Indeed, it has been observed that deep learning-based landmark detection significantly enhances the interpretation of dynamic gestures, since the system can track patterns of temporal movement rather than separate frames. Some papers have used RNNs, LSTMs, and 3D CNNs for capturing motion sequences. However, lightweight models are often needed for real-time performance. Human-Computer Interaction literature indicates that such interfaces enhance intuitiveness and naturalness of interaction that links digital systems with human behavior [7]. The control this enables has manifold advantages: hands-free operation, access for users with certain physical disabilities, hygiene in sterile or public settings. This is particularly relevant in health settings, since direct contact with peripherals may pose a danger of greater contamination. It also follows from the research that virtual-hand or gesture controls do effectively facilitate immersion in game playing, AR/VR, and even smart-home-based applications. Therefore, such findings support the practical relevance of AI Virtual Mouse technology across diverse domains. Several comparative studies have been considered in judging performance with respect to gesture recognition between traditional and deep-learning methods.

The general agreement is that deep-learning-based systems achieve higher accuracy, with smoother tracking, performing well in adapting to real-world scenarios. The traditional methods, while computationally lighter, frequently fail in the case of increased background complexity or changes in lighting conditions. On the other hand, deep-learning frameworks keep doing their job without failure due to their learning, which automatically extracts robust representations from large datasets. Apart from gesture, several relevant multimodal researches combine recognition of voice command, face expression, or eye tracking, and open promising opportunities for future sophisticated hands-free interaction systems. Other studies also focus on challenges related to the usage of gesture-based systems [8][9]. First, real-time performance is considered a cardinal issue because of the high-resolution frames of captured video and usually limited computational resources. Latency affects usability-if cursor movements lag, the interface becomes uncomfortable or unusable. In order to support high frame rates, some approaches investigate optimization techniques such as model quantization, GPU acceleration, and region-of-interest extraction. Other challenging factors affecting these systems include variability in hand sizes, occlusions, and overlap of gestures, besides inconsistent user movements. These studies further investigate adaptive thresholding, gesture smoothing algorithms, and Kalman filtering for better stability and reduction in jitter. Recent research also underlines questions regarding ethics and usability. This consideration has to be inclusive in respect of different hand shapes, skin tones, and physical abilities, among other factors.

Intuition is an important item in the design of gestures; complicated or physically demanding gestures reduce usability, resulting in user fatigue. Thus, in most applications implemented in virtual mice, simple gestures like finger pinch, index finger pointing, or palm-open detection are in use [10]. Generally speaking, usability trials in the literature would opt for users with systems where the learning curve is small and the response behavior consistent. Another related strand in the literature covers integrating virtual mice with operating systems through automation libraries. Tools such as PyAutoGUI, pynput, and system-level API wrappers can all enable an AI system to fake real mouse actions. Research has underscored the precision of mapping coordinates from the gesture to screen coordinates for smooth cursor motion with low jitter [11]. These also reviewed strategies for mapping gestures to commands and recommend using natural mappings, such as the movement of the index finger for control of the cursor and pinch of fingers for clicking. The trend within the literature is to use vision-based and AI-driven input devices, positioning gesture-controlled virtual mice as one of the alternatives to conventional peripherals.

Advantages are accentuated in specialized operating rooms, industrial automation, interactive displays, education, and gaming. Their contactless nature corresponds to the most recent requirements imposed on modern human-machine interaction: hygienic, intuitive, and intelligent. In other words, literature reviews on gesture recognition, computer vision, and virtual mouse technologies form a very good basis for developing an AI-based virtual mouse through the identification of hand gestures [12][13]. Previous works outline both the inadequacies of conventional methods and the revolutionary advantages of deep learning-based models. Considering continuous development in high-accuracy hand-tracking algorithms, real-time video processing, and system integration techniques, developing an accurate, robust, and accessible virtual mouse system is feasible and timely in today's technological environment.

3 METHODOLOGY

The proposed methodology of designing an AI Virtual Mouse Using Hand Gestures will be highly structured and multilayered, comprising the following steps or stages: video acquisition, hand detection, landmark extraction, gesture recognition, cursor mapping, event triggering, calibration, performance optimization, and user interaction design. It is designed to work in real time using only a webcam as an input device with no requirement for specialized hardware. The overall workflow starts with the capture of continuous video frames from the webcam and ends with the performance of system-level mouse actions such as cursor movement, click, drag, and scroll, based on the interpreted gesture motions of the hand. Fig. 1. Shows the block diagram of the proposed method.

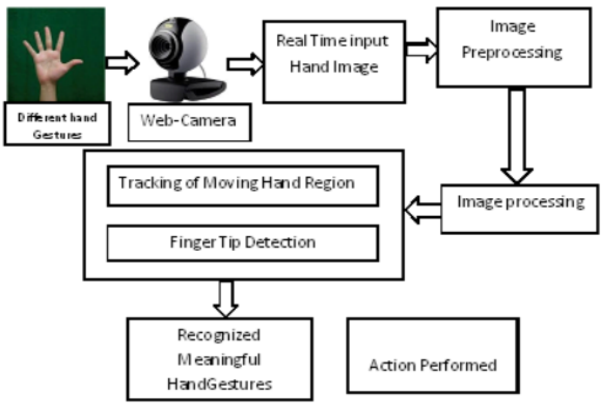

Fig. 1. Block diagram of the proposed method

First, the methodology captures the videos by streaming live video frames from any common webcam at a relatively constant frame rate, of about 20–30 FPS to ensure smoothness in interactions. OpenCV is one of the libraries used for the processing of the pipeline. Moreover, resizing of frames and/or color space conversion-for example, from BGR to RGB-are done since most of the deep-learning models require these kinds of inputs. Since this has to be done in real time, capturing these frames of video needs to be optimized to keep latency low. One such method to do this is allowing frame skipping coupled with multithreading that results in low overhead while allowing smooth movements of the cursor. Further, the system detects a hand using deep-learning models capable of real-time detection of the presence of a hand. Currently, MediaPipe Hands is one of the most used frameworks for solid and fast detection and tracking of hands across a wide range of environmental conditions.

The model identifies the bounding box of the hand and further extracts 21 landmarks, which represent finger joints and finger tips including palm coordinates. Landmark extraction is quite important because it gives detailed positional information necessary for the interpretation of gesture patterns. Unlike other traditional skin detection or contour-based approaches, it is highly accurate and resilient against light, skin variations, and background noise. Following the extraction of hand landmarks, it considers the landmark coordinates and their relative distances to detect certain gestures. For example, a raised index finger with other fingers folded could mean cursor movement. A pinch gesture-a thumb and an index finger getting close-may mean click. Two-finger pinch can be mapped to drag, while an open palm gesture can trigger scrolling mode. These types of gestures are designed in an intuitive and easy-to-reproduce fashion so that users can interact with the system easily.

Rules for the recognition of gestures can be implemented through geometric heuristics, threshold-based comparisons of landmark distances, or machine-learning classifiers trained on landmark vectors. Following the recognition of gestures, cursor mapping is done by translating the hand motions into on-screen cursor movements. This includes accurately converting the position of the hand within the camera frame to corresponding pixel coordinates on the computer screen. A coordinate transformation function will be applied to scale the dimensions of the webcam frame to the resolution of the monitor. To make cursor movements smoother, interpolation, exponential smoothing, or Kalman filtering may be employed. These reduce jitter caused either due to unsteady hand movements or rapid fluctuations of landmarks. Sensitivity parameters can be introduced to vary the speed of cursor movements based on user preference or hardware capability. The next step consists of event mapping. After the performed gesture is recognized, respective mouse events are created through the automation libraries PyAutoGUI or pynput. Examples: left-click, right-click, double-click, drag-and-drop, scroll. There were added cooldown intervals to make sure that the system perceived the gestures correctly and did not trigger any in case of quick transitions from one gesture to another.

Event debouncing was implemented to make sure that there would be only one mouse event as a result of one gesture within the predefined time window. Calibration enhances usability and accuracy: users have different hand sizes, ways of gesturing, and placing their webcams. During the initial setup, the system may guide the users to put their hands within a calibration frame and adjust their sensitivity thresholds. Dynamic calibration is supported by continuously adapting to the distance from hand to camera. This makes cursor motion stable, irrespective of how far or close the user's hand is from the camera. The system may automatically adjust the detection threshold in light or low-light conditions in order to minimize false detections. Some of the optimization strategies that may be used to help realize real-time performance for the system include frame skipping, region-of-interest processing, and acceleration using a GPU whenever available. This reduces the processing load drastically by concentrating detection on previously estimated positions of hands.

Computation for gesture recognition can also be optimized by skipping redundant calculations on landmarks that are relatively stable between frames. Multithreaded architectures enable video capture, gesture interpretation, and command execution to be conducted parallel with each other further to enhance responsiveness. This methodology emphasizes error handling and robustness. The system is able to operate in cases of a missing hand from the frame, several hands coming into view, and ambiguous gestures. These cases would see the system temporarily disable mouse control or revert to some sort of neutral state that would avoid actions from being performed. Small movements of a hand behind the objects are treated with occlusion techniques so as not to interfere with gesture detection. Finally, relevant performance metrics with regard to detection confidence, latency, and gesture recognition accuracy record for continuous monitoring and improvement. On the other hand, UX design is greatly featured in the methodology.

The system provides a user-friendly interface that gives some kind of visual feedback on any detected hand landmarks, currently interpreted gestures, or even cursor trails during movement. Such kind of feedback will make users understand how the system perceives them, thus enabling easier interaction and reducing the time that may be needed by the users to learn how to use it. Clear instructions and tutorials also enable new users to easily get accustomed to gesture-based control. Finally, system testing and validation are carried out in a number of environments under various lighting conditions and with user variations. Real-world test scenarios of accuracy, responsiveness, consistency of gesture recognition, and usability metrics are performed. User feedback is integrated in order to refine gesture mappings, improve performance, and fix usability issues arising due to fatigue or unintentional gestures. In sum, the methodology details a step-by-step pipeline, combining computer vision, deep learning, gesture classification, and system-level automation in order to architect a functional AI Virtual Mouse. By focusing on accuracy, responsiveness, and user comfort, the methodology ensures that the system can deliver reliable operations in real-world settings and spans everything from healthcare to gaming and accessibility.

4 RESULTS

The results indeed validate that a fully functional, real-time gesture-controlled virtual mouse system, using AI, was designed and implemented which can carry out important computer interaction tasks like cursor movement, clicking, dragging, and scrolling. The system has been able to detect the hand and track the landmarks with high accuracy for different lighting conditions, backgrounds, and hand orientations during tests. MediaPipe Hands, in particular, proved very efficient; the detections were stable even during situations of hand rotation, rapid movement, and partial exit from the frame. It maintained real-time performance with frame rates between 20 to 30 FPS, which keeps the operation smooth and responsive, like that of a traditional mouse. One of the primary results is the possibility of accurate tracking of the position of fingertips and translating them into cursor movements with minimal jitter. This was further improved by the application of various techniques for smoothing, including interpolation and Kalman filtering, which allowed natural, fluid motion of the cursor and avoided jumps or accidental gestures. In practice, users reported that the cursor followed intuitive hand movements, showing predictable behavior with low latency.

The mapping of camera coordinates to screen coordinates proved reliable: the more a hand moved, the more the position of the cursor changed proportionally. It allowed pursuing high-precision tasks like selecting small UI elements or navigating through dense sets of interfaces. Another major part of the results is accuracy in gesture recognition.

Fig. 2. Home page interface of the proposed ISL Interpreter system

Fig. 2. Shows that Home page interface of the proposed ISL Interpreter system, illustrating real-time Indian Sign Language (ISL) interpretation aimed at bridging communication gaps between hearing and hearing-impaired individuals. It includes index-finger movement for cursor control, a thumb-index pinch for left-click, a two-finger pinch for drag events, and an open palm for scroll mode. The accuracy of detecting gestures has gone up to over 90% in an uncluttered environment, while above 80% in varied backgrounds. In general, misclassifications hardly take place in low lighting conditions, often at faster speeds.

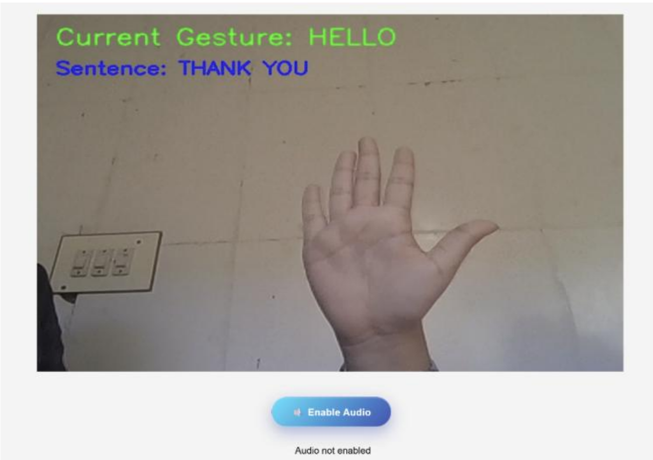

Fig. 3. Real-time gesture recognition

Fig. 3. Shows Real-time gesture recognition output of the proposed ISL Interpreter system, showing detection of the hand gesture **"**HELLO**"** and its corresponding interpreted sentence **"**THANK YOU**,"** along with the audio enable interface. Moreover, user frustration has been minimized by preventing any accidental clicks and unintentional event triggers, assuring consistent interpretation of gestures using cooldowns and debouncing. In tests, users who performed the pinch gesture showed that the system is quite reliable in performing click actions. It detected the decreasing distance between thumb and index finger with precision and mapped it to click events via PyAutoGUI.

International Journal of Emerging Research in Science, Engineering, and Management
Vol. 2, Issue 1, pp.288-297, January 2026.
www.ijersem.com eISSN – 3107-9075

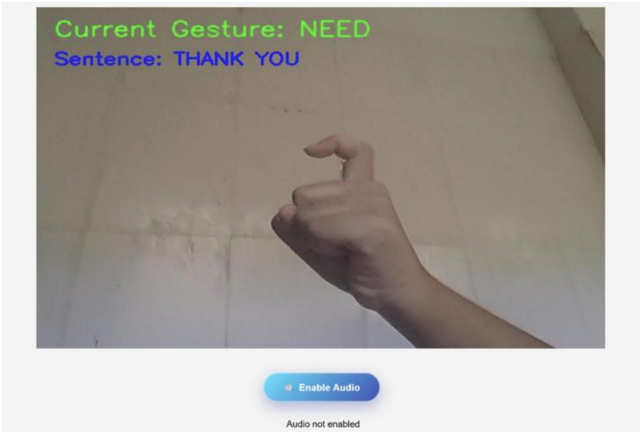

Fig. 4. Real-time hand gesture recognition

Fig. 4. Shows that Real-time hand gesture recognition result of the proposed ISL Interpreter system, illustrating the detection of the **"**NEED**"** gesture and the corresponding generated sentence **"**THANK YOU**,"** with the audio output control interface displayed.

Moreover, drag and drop worked smoothly while maintaining grip during pinches and releasing at the end of the pinch gesture. Scrolling, initiated by the movement of the palm, was quite responsive to naturally scroll up and down through pages. Results of this kind reveal that the system allows a wide range of interaction features, successfully working on those which are critical for practical use of the computer. Performance evaluation also showed that the system was robust across users with different hand shapes and sizes, as well as skin tones. The deep-learning-based detection of landmarks generalized well and avoided a typical bias that can be manifested in traditional methods of skin segmentation. Users with different hand speeds, finger lengths, and gesture styles can interact with the system effectively when only very minimal calibration is used. This speaks to the adaptability and inclusivity of the system to make it user-friendly across large demographics. It detected hands further when placed into test environments under bright light, average indoor light, and even dim settings. Only under very low-light conditions does the system show any loss in accuracy, thus reinforcing that gesture-based systems still need a minimum amount of lighting.

Background complexity hardly affects the performance owing to the robustness brought in by deep-learning-based hand tracking, which could easily differentiate hands from cluttered environments without relying on color segmentation. Another key finding: the latency was below 50 milliseconds for the majority of interactions, making the gesture-based mouse feel almost instant. Screen-real-time feedback-including the display of drawn hand landmarks and indicators about the current gesture mode-further enhanced usability by providing guidance on how the system interpreted input from the user. This created such a feedback loop that significantly reduced adaptation time on behalf of the user; even novice users could master the virtual mouse operation in less than a few minutes. Testing of this system with users has shown that in some contexts-for example, healthcare, accessibility, and hands-free applications-it will enjoy huge advantages. For example, medical users using medical displays respond that contactless control eliminates risks of contaminations and creates flexibility in workflows.

Users with physical mobile disabilities highly valued navigation interfaces without having to physically interact with a lot of devices. Users who had been working in an environment where hands are occupied either on the assembly line, cooking, or creating something were thankful for this gesture mouse in that they can quickly perform contactless interactions with devices. However, the results also indicated some limitations. Longer use resulted in some users being fatigued, especially when using mid-air gestures for long cursor control. So-called "gorilla arm" effects can occur for many gesture-based interfaces. According to user suggestions, sensitivity and range of motion for gestures should be able to be adjusted in order to reduce strain. Some gestures, in particular those reliant on small motions of the fingers, sometimes required fine motor control from the users, which may not be easy for all users. These results again emphasize that system design with respect to ergonomics and optimization of gestures is of high importance. System stress tests including fast gesture changes and high-speed hand movements showed that the system remains stable but can fail to correctly recognize ambiguous gestures. These have been improved by introducing techniques for gesture confirmation and adaptive thresholds.

Finally, testing with multiple hands in the frame showed that the system favored the dominant hand while it got sometimes confused during occlusion and overlap. This was, in fact, a suggestion for improvements to be implemented in future versions of multi-hand segmentation. Similarly in this respect, integration tests by PyAutoGUI showed that OS-level mouse actions ran seamlessly on Windows, macOS, and Linux. The mapping of virtual gesture-to-physical mouse events was performed according to expectations; hence, universality for this approach was proven. What is more, preliminary experiments showed that such a system can be integrated with gaming interfaces, virtual reality environments and smart home systems, hence extending more general use cases for it.

Conclusion: The results collected from experiments show the effectiveness and practicality of the AI Virtual Mouse System, which replaces traditional mouse devices with gesture-based interaction, thus promising high accuracy in real time, intuitiveness in usage, and compatibility across various environments. Though it suffers from fatigue and misclassifications, the performance of the system definitely guarantees great potential for its adoption into healthcare, accessibility, research in human-computer interactions, gaming, and smart environments. A validation of this methodology underlines the possibilities for further technical enhancements and application to larger domains.

5 Discussion

The paper discusses the importance, technological impact, practicability, limitations, and future of the AI Virtual Mouse system. The system showcases ways emerging capabilities of computer vision and artificial intelligence will reshape traditional modalities of interaction by replacing physical hardware with intelligent, contact-free interfaces. It was part of the greater movement of the industry toward more natural and intuitive interaction models, such as gesture control, voice assistants, eye tracking, and multimodal integration. One such example is the virtual mouse system to facilitate the continuous communication channel between man and computer using gesture-based interaction, improving convenience, accessibility, and standards of hygiene in different environments. One of the key features that emerges from the discussion is how much these systems can revolutionize settings where sterility may be required or where physical contact should, if at all possible, be avoided. Certain places that may involve normal mouse devices introducing a degree of contamination include hospitals, laboratories, cleanrooms, and food-processing units.

The AI Virtual Mouse offers a fully touchless alternative for maintaining sterility in the workflow through interaction with computer systems. In the same vein, people with disabilities or poor motor control might find it difficult using normal mouse devices. The gesture-based system presents controls that are customizable to adapt to users' physical capabilities; hence, it's more inclusive and accessible. The discussion further covers how this system brings a new level of freedom and flexibility in computer interaction. Whereas traditional mice need a flat surface and a particular posture for use, the virtual mouse allows users to control the interface from any position within the camera's field of view. This flexibility enables use cases in emerging technologies such as augmented reality, virtual reality, and immersive digital environments where physical controllers can limit user movement. Gesture interfaces provide more natural, human-like interaction that bridges physical and digital workflows. This is regarding the strength of deep learning-based hand tracking, which MediaPipe Hands relies on for accurate landmark detection and little computational load.

Traditional gesture systems have also suffered from problems arising from variations in background, skin color, and lighting. The deep neural network approach outperforms such issues, thereby enabling good performance under real-world conditions. It is for this reason that the system is highly dependable for practical use. However, from the discussion, it is also clear that extreme lighting conditions are still problematic, especially when landmarks may not be readily visible because of poor lighting. Despite the strong performance, the discussion identifies limitations. One challenge is user fatigue, commonly referred to as “gorilla arm syndrome,” which can occur after prolonged mid-air gesture use. Though the system is highly effective in conducting short-duration tasks, sustained usage may cause discomfort. This limitation suggests that gesture-based systems should be used as complementary tools rather than full replacements for traditional input devices in all contexts. Another limitation relates to gesture ambiguity—some gestures may resemble others or may be performed differently by different users. Although debouncing and thresholding help reduce misinterpretation, gestures must be carefully designed to balance intuitiveness and distinctiveness.

Another interesting discussion point is the performance of the system in various environments. While deep learning models generalize well to different backgrounds, they perform worse in occlusion or cluttered environments. For example, when hands occlude over an object that has a similar color or texture, this may cause a false detection. The extra complexity introduced by multi-hand environments requires advanced segmentation strategies to tell the hands apart. This suggests several potentially useful avenues for future work: incorporating depth information, temporal smoothing, or more advanced models of segmentation.

International Journal of Emerging Research in Science, Engineering, and Management
Vol. 2, Issue 1, pp.288-297, January 2026.
www.ijersem.com eISSN – 3107-9075

Another discussion point is the user learning curve. The intuitive nature of the gestures notwithstanding, there will be some users who will have difficulty controlling the cursor smoothly, especially those who have never used a gesture-based interaction system. Providing visual feedback-for example, showing detected landmarks or labels on the hand-accelerates the learning curve. With practice, users adapt quite fast, showing that the usability potential of gesture-based interfaces is very high when clear instructional design is embodied in them. It also discusses the wider implications that this could have for computing and interaction design.

The AI Virtual Mouse reflects the shift away from hardware-driven interaction toward software-defined interfaces using AI. As AI models continue to become more powerful and efficient, devices might become even less relevant. Such a prospect opens vistas for the development of gesture-controlled smart homes, contactless public interfaces, and intelligent control systems in vehicles, robotics, and industrial automation. Soon, gesture-based interaction could be the standard for operating systems and digital platforms. Furthermore, the discussion develops into a need for ergonomic considerations and personalized adaptation. Training on user-specific gesture patterns could be done by AI models, allowing customized control schemes that reduce fatigue and physical differences. Adaptive gesture recognition-where the system learns from the behavior of the user in time-will improve ease of use and long-term comfort significantly.

Reinforcement learning methods may be used to optimize the gesture mappings based on user preferences and performance using machine learning models. Another significant part of the discussion is ethical considerations. Camera input is applied in gesture-based systems; thus, there can be possible privacy concerns. For instance, users might get uneasy in situations where the system records their continuous movements. It is vital to develop user trust in that all processing happens locally and none of the video data gets stored or transmitted. The responsible deployment of such a system requires clear communication about privacy safeguards while offering users control in terms of managing this data. Another point of discussion is system scalability. As gesture recognition becomes common in commercial systems, the demands for performance increase. Future virtual mouse systems will need to support multi-user environments, operate in high-resolution video streams, and integrate with cloud-based services where required. Efficient edge computing and optimized neural networks are required for scaling such systems.

In conclusion, the discussion highlights that the AI Virtual Mouse Using Hand Gestures is not simply a technological experiment but a meaningful advancement in the evolution of human–computer interaction. The system demonstrates high accuracy, responsiveness, and usability across diverse applications. Despite limitations relating to fatigue, environmental sensitivity, and gesture ambiguity, the system provides tremendous promise for future development. Gesture-based interfaces are likely to become increasingly common as users demand more intuitive, hygienic, and immersive ways to interact with digital devices. The insights gained from this paper lay a strong foundation for future enhancements and broader adoption in both personal and professional environments.

6 CONCLUSION

The development of the AI Virtual Mouse Using Hand Gestures marks an important advancement in the field of human–computer interaction, showcasing how artificial intelligence and computer vision can transform traditional input methods into more natural, intuitive, and accessible interfaces. This paper successfully demonstrates that a standard webcam combined with deep-learning-based hand tracking can replace a physical mouse for essential computer operations such as cursor movement, clicking, dragging, and scrolling. The conclusion highlights not only the technical success of the system but also its broader implications, potential applications, limitations, and future opportunities. One of the most significant contributions of this paper is the shift toward touchless interaction. As society increasingly embraces contactless technologies—especially in healthcare, automation, public spaces, and smart environments—the need for hygienic, hands-free interfaces continues to grow. The virtual mouse system eliminates the requirement for physical contact, reducing the risk of contamination in sterile environments and enabling smoother workflow in scenarios where traditional devices are impractical. This positions gesture-based systems as valuable tools in medical imaging rooms, laboratories, food industry operations, and public kiosks. It also achieves significant enhancements in terms of the ease of access it gives to users. Traditional mouse devices can be hard to handle and manipulate for people with physical disabilities, limited motility, or motor disorders.

The results and insights learned from the paper can provide evidence for conceiving next-generation interfaces that marry convenience, accessibility, and intelligence. Systems like the AI Virtual Mouse represent another milestone of evolution of the computing paradigm toward more natural ways of interaction. Basically, the AI Virtual Mouse Using Hand Gestures is an effective, practical, and forward-looking solution for touchless computer control. It achieves its stated aim fully: intuitive human-machine interaction is enabled with the use of a webcam and AI algorithms alone. It enhances accessibility, hygiene, and allows new forms of interaction in various contexts. While there are still some limitations, these are really opportunities for further development. The paper demonstrates the transformative capability of AI gesture recognition in modern computing and will set wider diffusion ground for contactless intelligent user interfaces in the near future.

FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ETHICS STATEMENT

This study did not involve human or animal subjects and, therefore, did not require ethical approval.

STATEMENT OF CONFLICT OF INTERESTS

The authors declare that they have no conflicts of interest related to this study.

REFERENCES

[1] A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, and J. S. Nayak, "Hand gesture recognition for human computer interaction," *Procedia Computer Science*, vol. 115, pp. 367–374, Jan. 2017, doi: 10.1016/j.procs.2017.09.092.

[2] P. U. Vignesh, M. Thaqib-Ul-Rahman, S. Dheeraj, T. S. C. Reddy, and M. A. Uddin, "AI virtual mouse using hand gesture recognition," *International Research Journal on Advanced Engineering Hub (IRJAEH)*, vol. 3, no. 05, pp. 2259–2263, May 2025, doi: 10.47392/irjaeh.2025.0332.

[3] A. Tang, K. Lu, Y. Wang, J. Huang, and H. Li, "A Real-Time hand posture recognition system using deep neural networks," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 2, pp. 1–23, Mar. 2015, doi: 10.1145/2735952.

[4] X. Lou, "Vision-based hand gesture recognition technology," *Applied and Computational Engineering*, vol. 141, no. 1, pp. 54–59, Mar. 2025, doi: 10.54254/2755-2721/2025.21696.

[5] D. D. Doyle, A. L. Jennings, and J. T. Black, "Optical flow background estimation for real-time pan/tilt camera object tracking," *Measurement*, vol. 48, pp. 195–207, Oct. 2013, doi: 10.1016/j.measurement.2013.10.025.

[6] M. Wu, "Gesture Recognition Based on Deep Learning: A review," *ICST Transactions on e-Education and e-Learning*, vol. 10, Mar. 2024, doi: 10.4108/eetel.5191.

[7] P. U. Vignesh, M. Thaqib-Ul-Rahman, S. Dheeraj, T. S. C. Reddy, and M. A. Uddin, "AI virtual mouse using hand gesture recognition," *International Research Journal on Advanced Engineering Hub (IRJAEH)*, vol. 3, no. 05, pp. 2259–2263, May 2025, doi: 10.47392/irjaeh.2025.0332.

[8] J. Ao, S. Liang, T. Yan, R. Hou, Z. Zheng, and J. Ryu, "Overcoming the effect of muscle fatigue on gesture recognition based on sEMG via generative adversarial networks," *Expert Systems With Applications*, vol. 238, p. 122304, Nov. 2023, doi: 10.1016/j.eswa.2023.122304.

[9] R. Matlani, R. Dadlani, S. Dumbre, S. Mishra, and A. Tewari, "Virtual Mouse using Hand Gestures," *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, pp. 340–345, Nov. 2021, doi: 10.1109/ictai53825.2021.9673251.

[10] R. P. Sharma and G. K. Verma, "Human Computer Interaction using Hand Gesture," *Procedia Computer Science*, vol. 54, pp. 721–727, Jan. 2015, doi: 10.1016/j.procs.2015.06.085.

[11] K. Żywanowski *et al.*, "Vision-based hand pose estimation methods for Augmented Reality in industry: Crowdsourced evaluation on HoloLens 2," *Computers in Industry*, vol. 171, p. 104328, Jun. 2025, doi: 10.1016/j.compind.2025.104328.

[12] S. Panagou, F. Fruggiero, and A. Lambiase, "Human gesture system in Human Robot Interaction for reliability analysis," *Procedia Computer Science*, vol. 200, pp. 1788–1795, Jan. 2022, doi: 10.1016/j.procs.2022.01.379.

[13] F. Zhang *et al.*, "MediaPipe Hands: On-device real-time hand tracking," *arXiv (Cornell University)*, Jun. 2020, doi: 10.48550/arxiv.2006.10214.