Gesture Controlled Virtual Mouse Using Computer Vision

Manikanta Gugulothu[1], Meenakshi Sakre[2], Rakshitha Vennamaneni[3], Raheem Mohammad[4], Mr. M. RamaMohan[5]

[1-4] Undergraduate Students, Department of Information Technology, Vidya Jyothi Institute of Technology, Hyderabad, Telangana, India
[5] Assistant Professor, Department of Information Technology, Vidya Jyothi Institute of Technology, Hyderabad, Telangana, India

---***---

Abstract - *This paper presents the development of a gesture-controlled virtual mouse using computer vision techniques. The system enables users to control a computer cursor using hand gestures captured through a standard webcam, eliminating the need for a physical mouse. The proposed approach uses MediaPipe for hand landmark detection and OpenCV for real-time video processing. By tracking 21 key points on the hand, the system interprets different gestures and maps them to common mouse operations such as cursor movement, clicking, dragging, and scrolling. During development, emphasis was placed on achieving smooth cursor control and maintaining reliable performance under normal lighting conditions. The system provides a simple and contactless method of interaction, making it useful in environments where touch-free operation is preferred. In addition to improving user convenience, the system can also assist individuals with physical limitations. Since it requires only a webcam and no additional hardware, it offers a cost-effective alternative to traditional input devices. Overall, the proposed system demonstrates a practical approach to gesture-based human-computer interaction.*

Key Words: **Gesture Recognition, Computer Vision, Virtual Mouse, Human-Computer Interaction, MediaPipe, OpenCV, Hand Tracking, Touchless Interaction**

1.INTRODUCTION

In recent years, touch-free interaction has gained significant attention in the field of modern computing. Instead of relying only on traditional input devices such as a mouse or keyboard, there is a growing interest in interacting with systems using natural human gestures. With the increasing demand for intuitive and user-friendly interfaces, gesture-based systems are becoming an important area of research in human-computer interaction. Although conventional input devices are widely used, they have certain limitations. In environments such as hospitals or shared workspaces, continuous physical contact with devices can raise hygiene concerns. In addition, people with physical disabilities may find it difficult to use standard input hardware. These challenges motivated us to explore a touchless interaction approach using computer vision. In this work, we focus on developing a virtual mouse that can be controlled using hand gestures. The idea is to use a webcam to capture real-time hand movements and convert them into cursor actions such as movement, clicking, dragging, and scrolling. This provides a more natural way of interacting with the computer without requiring any physical device. To implement this system, we used MediaPipe and OpenCV. MediaPipe is used for detecting and tracking hand landmarks, while OpenCV handles image processing and real-time video analysis. During development, we found that combining these two frameworks provides a good balance between accuracy and performance under normal conditions. Apart from improving user convenience, touch-free systems can also enhance accessibility and maintain better hygiene in certain environments. Such systems can be useful in applications like virtual reality, gaming, assistive technologies, and smart environments. Overall, this project demonstrates a simple and cost-effective approach to building a gesture-controlled virtual mouse using computer vision, making human-computer interaction more natural and flexible.

2. LITERATURE SURVEY

[1] Kumar et al. proposed a hand gesture recognition system using Convolutional Neural Networks (CNNs) for classifying static gestures such as open palm, fist, and pointing. The model achieved high accuracy due to strong feature extraction capabilities of CNNs. However, the system requires high computational resources, including GPU support, and struggles with dynamic gestures and occlusion scenarios.

[2] Patel et al. introduced a real-time hand tracking system using MediaPipe, which detects 21 hand landmarks for gesture recognition. The system performs efficiently even on low-end devices and supports real-time applications. However, its accuracy decreases under poor lighting conditions and complex backgrounds.

[3] Patel et al. presented a vision-based hand gesture recognition system using OpenCV techniques such as contour detection, background subtraction, and convex hull analysis. The approach is simple and suitable for real-time applications with low computational cost. However, it is sensitive to noise and lighting variations, resulting in limited accuracy.

[4] Singh et al. developed a real-time gesture-based virtual mouse system using MediaPipe for hand tracking and rule-based gesture mapping for mouse operations such as cursor movement and clicking. The system provides lightweight and real-time performance without requiring additional hardware. However, it supports a limited set of gestures and shows reduced accuracy during fast hand movements.
[5] Kumar et al. proposed a hand gesture recognition system using machine learning classifiers such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN). The system uses extracted hand features like finger distances and angles for classification, achieving better accuracy than traditional methods. However, it requires a large amount of training data and involves higher computational cost. Despite these advancements, existing systems face several challenges such as sensitivity to lighting conditions, reduced performance in dynamic environments, limited gesture sets, and dependency on high computational resources. To overcome these limitations, the proposed system integrates MediaPipe and OpenCV to achieve accurate, real-time gesture recognition with minimal hardware requirements, enabling efficient cursor control and improved user interaction. To address these challenges, the proposed system introduces a gesture-controlled virtual mouse using computer vision techniques that combines the efficiency of MediaPipe for hand tracking with the flexibility of OpenCV for real-time processing. The system is designed to perform cursor movement and mouse operations such as clicking, dragging, and scrolling using simple hand gestures, ensuring a balance between accuracy, performance, and cost-effectiveness.

3.PROPOSED MODEL

The proposed system is a gesture-controlled virtual mouse that enables users to interact with a computer without using a physical mouse. In this work, we use a standard webcam to capture real-time video input, which is then processed using computer vision techniques to interpret hand gestures.
The system continuously captures video frames through the webcam. These frames are processed using the MediaPipe framework to detect and track hand landmarks. MediaPipe provides 21 key points on the hand, which makes it possible to accurately track finger positions and movements in real time.
Based on these landmarks, different gestures are recognized using simple predefined conditions. OpenCV is used for image processing and handling video frames, which helps in maintaining smooth and efficient real-time performance.
The detected hand movements are mapped to screen coordinates to control the cursor. Different gestures are assigned to perform mouse operations such as cursor movement, left click, right click, dragging, and scrolling. For instance, the movement of the index finger is used to control the cursor, while specific combinations of fingers are used to perform click actions.

During implementation, we observed that small hand movements could cause unwanted cursor fluctuations. To address this, smoothing and thresholding techniques were applied to improve cursor stability and reduce noise. The gesture recognition logic was also designed to work reasonably well under moderate lighting variations.
The overall system is divided into multiple stages, including video acquisition, hand detection, landmark extraction, gesture recognition, and action mapping. This modular approach makes the system easier to understand and allows future improvements.
The proposed model is cost-effective and does not require any additional hardware apart from a webcam. It can be useful in applications where touch-free interaction is preferred and can also assist users with physical limitations. With further improvements, the system can be extended to areas such as virtual reality, gaming, and smart environments.

4.METHODOLOGY

The proposed gesture-controlled virtual mouse system follows a structured workflow consisting of five main stages: video acquisition, hand detection, landmark extraction, gesture recognition and mapping, and cursor control. This step-by-step approach helps in achieving accurate and real-time touch-free interaction with the computer.

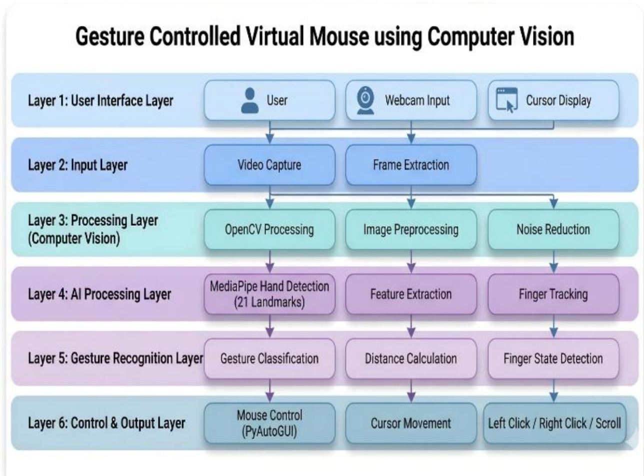

Fig -1: System Architecture

4.1 Video Acquisition

The process begins with capturing real-time video input using a standard webcam. The camera continuously captures frames, which are then processed by the system. These frames provide the necessary input for detecting and tracking hand movements.

4.2 Hand Detection

In this stage, MediaPipe is used to detect the presence of a hand in each video frame. The system identifies the hand region and separates it from the background. During testing, we observed that the detection works well under normal lighting conditions, although performance slightly decreases in low-light environments.

4.3 Landmark Extraction

Once the hand is detected, MediaPipe extracts 21 key landmark points representing different parts of the hand, including finger joints and tips. These landmarks provide precise positional information, which is used to understand finger movement in real time.

4.4 Gesture Recognition and Mapping

The system uses predefined rules to recognize different hand gestures and map them to mouse operations. An open palm gesture is used to initialize the system. Cursor movement is controlled using the index finger, while different finger combinations are used for actions such as clicking and dragging. For example, bringing the index and middle fingers close together triggers a click event, while a closed hand gesture is used for dragging. These gestures were selected based on trial and testing to ensure better reliability during usage.

4.5 Cursor Control and Action Mapping

In the final stage, the recognized gestures are converted into corresponding mouse actions. The movement of the index finger controls the cursor position on the screen. Different gestures trigger operations such as left click, right click, dragging, and scrolling. OpenCV is used for screen mapping and to ensure smooth cursor movement. During implementation, smoothing techniques were applied to reduce jitter and improve overall user experience.

5.RESULT AND DISCUSSION

The proposed gesture-controlled virtual mouse system was developed using Python and implemented with the help of OpenCV, MediaPipe, and PyAutoGUI libraries. The system was tested in real time under different conditions to evaluate its performance, reliability, and usability. During testing, we considered key performance factors such as accuracy, response time, frame rate, and overall system stability. We observed that the system performs consistently across most gestures and provides smooth cursor control during normal usage. Under standard lighting conditions, the system achieved an accuracy of around 94% to 96%. Among all the gestures, cursor movement showed the highest accuracy, mainly because it relies on continuous tracking of finger landmarks. On the other hand, gestures like dragging and scrolling showed slightly lower accuracy, as they depend on multiple finger positions and their coordination. In terms of responsiveness, the system showed an average response time between 25 ms and 45 ms, which allows near real-time interaction without noticeable delay. The frame rate was observed to be between 22 and 28 frames per second, providing smooth visual feedback and stable cursor movement. We also noticed that system performance depends on environmental conditions. It works best in well-lit environments with moderate hand movement, while low lighting and very fast gestures can slightly reduce accuracy.

Table -1: Performance Evaluation of Gesture Operations

Gesture	Accuracy(%)	Response Time(ms)
Cursor Movement	96	25
Left Click	95	35
Right Click	94	40
Drag	93	45
Scrolling	92	45

The system performed well under different conditions but showed slight accuracy reduction in low light. Moderate hand movement produced the best results.

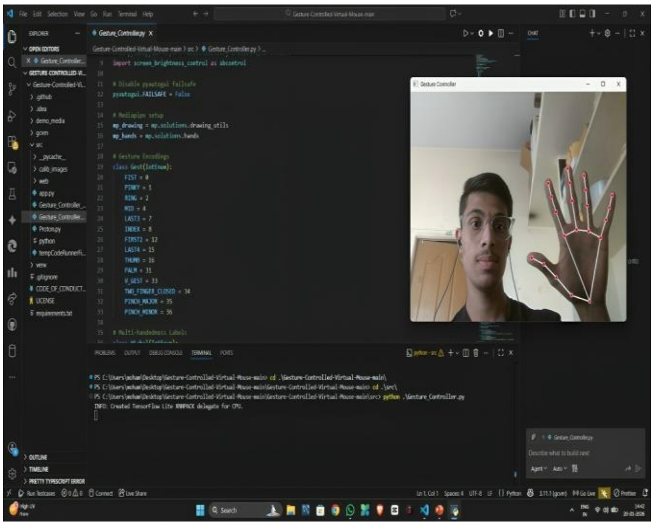

Fig -2: Initialization of Hand Detection

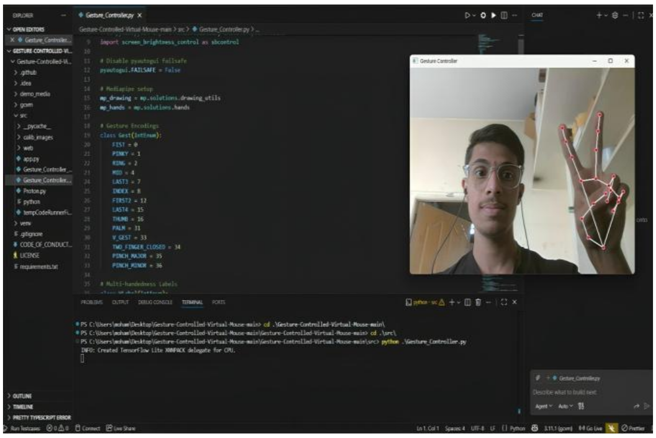

Fig -3: Cursor Movement

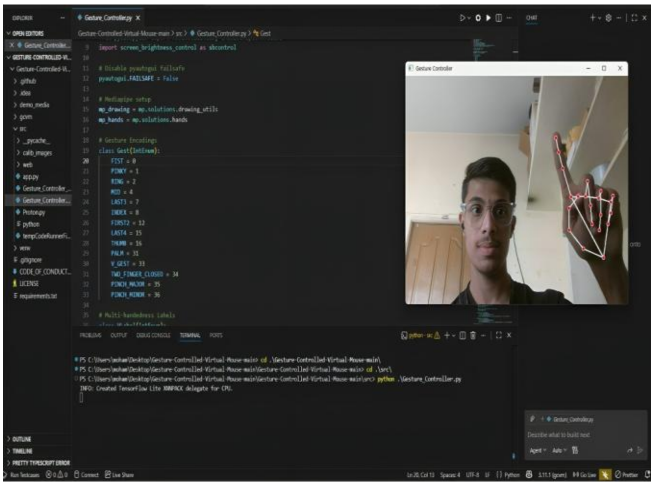

Fig -4: Left Click

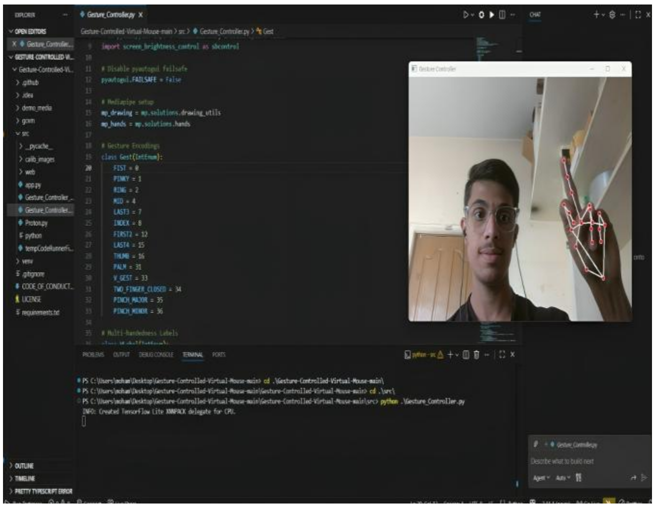

Fig -5: Right Click

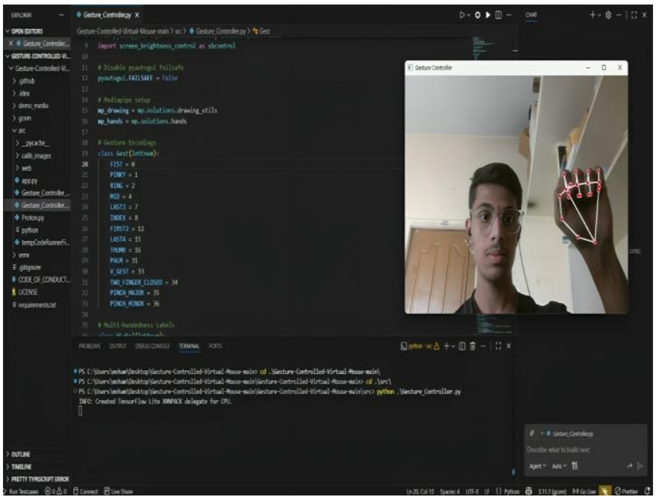

Fig -6: Drag

The initialization and hand detection process is shown in Fig -2. Cursor movement using hand gestures is illustrated in Fig -3. The left click operation is demonstrated in Fig -4, while the right click operation is shown in Fig -5. Drag operation using gestures is presented in Fig -6. These figures confirm the practical implementation of the system and show its real-time functionality. During testing, we observed that environmental conditions have a noticeable impact on system performance. The system works best in well-lit environments with minimal background interference, where hand landmarks are clearly detected. In low-light conditions or cluttered backgrounds, a slight drop in accuracy was observed due to difficulty in detecting precise hand features. User behaviour also affects performance. Moderate and controlled hand movements result in smoother interaction and better accuracy. In contrast, very fast or abrupt gestures can introduce minor delays due to processing limitations. When compared to traditional input devices, the proposed system offers a flexible and contactless way of interaction without requiring additional hardware. Although a physical mouse provides slightly higher precision, the proposed system performs well for general-purpose tasks while offering advantages such as improved hygiene and accessibility. Overall, the results indicate that the system is efficient, responsive, and suitable for real-time applications. The combination of MediaPipe and OpenCV provides reliable gesture recognition while keeping computational requirements low, making the system practical for everyday use.

6.CONCLUSION

This work presents a practical approach for enabling touch-free interaction with a computer using hand gestures. In this project, we developed a system that allows users to control cursor movement and perform mouse operations using a standard webcam, without relying on traditional input devices. This makes the interaction more intuitive and user-friendly. The system integrates MediaPipe for hand landmark detection and OpenCV for real-time processing, which helps in accurately tracking hand movements. Different gestures are mapped to operations such as cursor movement, clicking, dragging, and scrolling. The use of finger position and distance-based calculations provides reliable gesture recognition while maintaining real-time performance. From the experimental results, we observed that the system performs well under normal conditions, providing good accuracy with minimal delay. It works best in well-lit environments with moderate hand movements and does not require any additional hardware, making it accessible to a wide range of users. The proposed system reduces dependency on physical input devices and offers a hygienic and contactless method of interaction. It also adapts reasonably well to different users without requiring complex calibration. However, some limitations were identified during testing. The system shows reduced performance in low-light conditions and during very fast hand movements. In addition, complex backgrounds can affect detection accuracy. These challenges highlight the need for further improvements in robustness. Overall, the system demonstrates a practical alternative to traditional mouse-based interaction and shows the potential of gesture-based interfaces in modern computing. With further enhancements, it can be extended to applications such as virtual reality, gaming, smart systems, and assistive technologies.

7.FUTURE SCOPE

The current system provides a good foundation for touch-free human-computer interaction, but there are several areas where further improvements can be made. One important area is improving system performance under different environmental conditions. During testing, we observed that low lighting and complex backgrounds can affect detection accuracy. Future work can focus on enhancing robustness using improved image processing techniques. Another possible improvement is the integration of machine learning or deep learning models for gesture recognition. This can help the system adapt to different users and support a wider range of dynamic and personalized gestures. The system can also be extended to support multi-hand detection and more complex gesture combinations, which would enable advanced functionalities. In addition, optimizing the system for higher frame rates and lower latency can further improve real-time performance, especially for fast hand movements. The proposed system also has strong potential applications in the healthcare domain, where touch-free interaction can help reduce contamination risks. It can also be integrated with virtual reality and augmented reality systems to provide more immersive interaction. Furthermore, the system can be applied in areas such as gaming, robotics, and assistive technologies for individuals with physical disabilities. In future, deploying the system on mobile and embedded platforms can improve portability and make it more accessible. With these enhancements, the system can evolve into a more efficient and scalable solution for next-generation human-computer interaction.

8.REFERENCES

[1] N. Sharma and A. Gupta, "A Real Time Air Mouse Using Video Processing," International Journal of Advanced Science and Technology, vol. 29, pp. 4635–4646, 2020.

[2] P. Mishra and K. Sarawadekar, "Fingertips Detection in Egocentric Video Frames Using Deep Neural Networks," in Proc. International Conference on Image and Vision Computing New Zealand (IVCNZ), 2019, pp. 1–6.

[3] S. Shriram, B. Nagaraj, J. Jaya, S. Shankar, and P. Ajay, "Deep Learning Based Real Time AI Virtual Mouse System Using Computer Vision," Journal of Healthcare Engineering, 2021.

[4]"Virtual Mouse Using MediaPipe and OpenCV," International Journal of Research Publication and Reviews, 2024.

[5] "Real Time Hand Gesture Controlled Virtual Mouse using MediaPipe," International Journal of Advanced Science and Research Management, 2023.

[6] R. Matlani, R. Dadlani, S. Dumbre, S. Mishra, and A. Tewari, "Virtual Mouse using Hand Gestures," in Proc. International Conference on Technological Advancements and Innovations (ICTAI), 2021, pp. 340–345.

[7]"Gesture Controlled Virtual Mouse with Voice Automation," International Journal of Engineering Research and Technology, 2023.

[8] T. Grzejszczak, R. Molle, and R. Roth, "Tracking of Dynamic Gesture Fingertips Position in Video Sequence," Archives of Control Sciences, 2020.

[9] A. Kumar, A. Chaudhary, A. Singhal, A. Dev, and A. Jaiswal, "Gesture and Voice Controlled Virtual Mouse: A Review," in Proc. International Conference on Disruptive Technologies (ICDT), 2024, pp. 876–879.

[10] G. Sung et al., "On Device Real Time Hand Gesture Recognition," arXiv, 2021.

[11] A. Morajkar, A. M. James, M. Bagwe, A. S. James, and A. Pavate, "Hand Gesture and Voice Controlled Mouse," Advanced Engineering Days, vol. 6, pp. 127–131, 2023.

[12] S. R. Chowdhury, S. Pathak, and M. D. A. Praveena, "Gesture Recognition Based Virtual Mouse and Keyboard," in

Proc. International Conference on Trends in Electronics and Informatics (ICOEI), 2020, pp. 585–589.

[13] V. V. Reddy, T. Dhyanchand, G. V. Krishna, and S. Maheshwaram, "Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition," in Proc. IEEE HYDCON, 2020, pp. 1–5.

[14] R. Titlee, A. U. Rahman, H. U. Zaman, and H. A. Rahman, "A Novel Design of an Intangible Hand Gesture Controlled Computer Mouse Using Vision Based Image Processing," in Proc. International Conference on Electrical Information and Communication Technology (EICT), 2017, pp. 1–4.

[15] F. Zhang et al., "MediaPipe Hands: On Device Real Time Hand Tracking," arXiv, 2020.

[16] K. H. Shibly, S. K. Dey, M. A. Islam, and S. I. Showrav, "Design and Development of Hand Gesture Based Virtual Mouse," in Proc. International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019, pp. 1–5.

[17] J. Kotti, B. Padmaja, and D. Deepa, "Enhancing Gesture Controlled Virtual Mouse and Virtual Keyboard Using AI Techniques," Journal of Mobile Multimedia, vol. 20, no. 2, pp. 473–493, 2024.

[18] V. V. Reddy, T. Dhyanchand, G. V. Krishna, and S. Maheshwaram, "Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition," IEEE HYDCON, 2020.

[19] D. Kalaivani et al., "Enhancing Accessibility Through Gesture Based Human Computer Interaction: A Virtual Mouse Approach," in Proc. ICDSMLA, Lecture Notes in Electrical Engineering, vol. 1528, Springer, 2026.

[20] D. Sanka et al., "AI Virtual Mouse: Revolutionizing Human Computer Interaction," in Intelligent Communication, Control and Devices, Lecture Notes in Networks and Systems, vol. 1164, Springer, 2025.