International Journal of Advanced Research and Review

www.ijarr.org

Hand Motions Based Virtual AI Mouse

[1] A. Dileep,[2] Gayathri Vaidya,[3] Sambari Eshwari Vani,[4] Mangalagiri Uday Kiran,[5] Temburu Himagiri,
[1]Assistant Professor, Department of ECE, Narsimha Reddy Engineering Collage, Maisammaguda(V), Kompally, Telangana.
[2,3,4,5] Student, Department of ECE, Narsimha Reddy Engineering Collage, Maisammaguda(V), Kompally, Telangana.

Abstract— Despite touchscreens' monopoly on portable devices in the modern digital world, they remain prohibitively pricey for PC workstation systems. However, a tacit human-computer junction, most notably a mouse controlled by hand gestures, may bridge this gap.This study proposes a novel smart-mouse program that uses a camera to detect and understand user hand gestures for system control.In order to do actions like as streeling, left-clicking, right-clicking, and cursor helm without requiring a physical device, the system records and processes hand gestures using OpenCV and advanced machine learning simulations. List of In this context, "terms" refer to things like object tracking, OpenCV, Mediapipe, palm detection model, hand gesture recognition, and human-computer interface.

I. INTRODUCTION

Because of how fast computers are becoming, Human-Computer Interaction (HCI) has become more important than ever before [21]. Touchscreens may be ubiquitous on mobile devices, but desktop computers still can't justify the expense or inconvenience [22]. A good alternative in this case would be virtual interaction devices powered by computer vision. A virtual artificial intelligence mouse that mimics the feel and movement of a user's hands offers a natural and effortless alternative to conventional input devices like a trackpad or mousev[23]. In this research, we show an AI-driven method that uses a camera, MediaPipe, OpenCV, and complex algorithms to detect and understand hand motions [24]. You can control the cursor, scroll, click left and right, drag, change the brightness, and even transmit files via socket programming using fingertip detection and gesture endorsement [25]. Improved accessibility and intuitiveness of interaction are achieved via the use of affordable hardware and computer vision algorithms in this system.
Computer games, augmented reality, biomedical devices, and assistance for those with physical impairments are just a few of the many potential uses for gesture recognition beyond only enabling natural interaction with technology [26]. Following a four-step approach that includes picture preprocessing, region extraction, feature extraction, and feature matching, the system is able to provide accurate and immediate gesture identification. The use of hand gestures to control computers is fraught with its own set of difficulties. Not all users can utilize conventional input methods, such as physical trackpads and mice, particularly those with mobility issues. While gesture-based controls provide a more organic and expressive alternative, the existing approaches are either too expensive or too inaccurate for widespread use. This study use inexpensive cameras to build a real-time hand motion detection system in order to circumvent these problems. The technology eliminates the need for physical input devices by using hand gestures, a common and natural method of communication. Using hand motion detection for spoken scientific and other interaction understanding, the endeavor aims to produce accessible, user-friendly software for desktops and laptops.

II. LITERATURE REVIEW

A literature review synthesizes and summarizes all of the previous research on a certain subject. It offers an analysis and critique of previous studies, hypotheses, and research methods that have been published on a certain subject or field. Table -1 provides a comprehensive analysis of the various modalities discussed in the research, including their advantages, disadvantages, reliability, and practical applications. Each modality represents a distinct approach or technical development in the context of virtual AI mouse systems and related disciplines. By outlining the pros, cons, and practical uses of each method, the comparison aids in selecting or developing the most effective ways to gesture-based HCI. The citations are supplied to assist with future research and assessment of each method.

Table 1. Comparison of Modalities in Virtual AI Mouse Systems Using Hand Gestures

Citation	Modality	Advantages	Disadvantages	Reliability	Application
[1]	Speech Recognition	High accuracy in controlled environments; hands-free interaction	Limited in noisy environments; language-dependent	Moderate to High	Virtual assistants, accessibility tools
[2]	Gesture Recognition	Intuitive interaction; suitable for diverse use cases	Requires specialized hardware; computationally intensive	High in controlled settings	Gaming, robotics, medical systems
[3]	Vision-Based Systems	No physical contact required; supports complex interactions	Affected by lighting conditions; high computational demand	Moderate	Surveillance, AR/VR systems, automotive interfaces
[4]	Dataglove-Based Input	High accuracy for finger-specific gestures	Expensive; cumbersome to wear	High	Robotics, virtual reality, sign language recognition
[5]	Kinect Depth Cameras	Accurate depth sensing; works in low light	High cost; limited range	Moderate to High	Gesture recognition, skeleton tracking, accessibility systems
[6]	Color Detection Systems	Low-cost hardware; easy implementation	Limited robustness in dynamic environments	Moderate	Object tracking, interaction systems
[7]	Infrared Sensor Systems	Effective in low-light conditions; robust detection	Limited range; subject to interference	High	Gesture-based gaming, industrial automation
[8]	Deep Learning-Based Methods	High accuracy with large datasets; adaptable to complex tasks	Requires extensive training data; computationally expensive	High	Autonomous vehicles, medical diagnostics, advanced HCI systems
[9]	RGB-D Image Recognition	Combines visual and depth data for enhanced accuracy	Hardware-specific; higher processing requirements	High	Sign language recognition, motion tracking
[10]	FPGA-Based Architectures	High speed; energy-efficient	Specialized expertise required; design complexity	High	Real-time systems, video processing, gesture recognition
[11]	Optical Mouse Sensors	High precision tracking; cost-effective	Limited to specific surface types	Moderate	HCI input devices, low-cost tracking systems
[12]	U-Net Architectures	High segmentation accuracy	Limited interpretability; computationally heavy	High	Medical imaging, object detection
[13]	IoT Systems	Remote monitoring and control; scalable	Security concerns; reliance on network connectivity	Moderate to High	Smart homes, industrial automation
[14]	Skeleton Gesture Recognition	High precision for movement tracking; robust recognition under controlled environments	Limited range; dependent on sensor quality	High	Robotics, interactive systems, gaming
[15]	Interactive Car Games	Intuitive gameplay; integrates gesture-based interaction	Limited gesture set; requires specific setups	Moderate	Gaming, interactive systems
[16]	ASL Recognition Systems	Facilitates communication for hearing-impaired; leverages depth data	Limited dataset diversity; complex preprocessing	High	Accessibility tools, communication aids
[17]	Garbage Accumulation Robots	Automates waste collection; reduces manual intervention	Dependent on IoT infrastructure; subject to network failures	Moderate	Smart cities, industrial waste management
[18]	Kogge-Stone Adders	Low latency; efficient for high-performance computations	Requires CMOS design expertise	High	High-speed processing, computational architectures

[19]	Glove-Based Input Systems	High accuracy in tracking finger movements	High cost; cumbersome for extended use	Moderate	Virtual reality, robotics, training simulators
[20]	ASL RGB-D Alignment	Combines RGB and depth for accurate ASL recognition	Hardware-specific solutions; higher processing power required	Hardware-specific solutions; higher processing power required	Sign language translation systems

Fig. 1. Distribution of References by modality

Figure 1 is a pie chart showing the distribution of references across various publications and conference proceedings. The inclusion of a diverse range of sources in the important results implies that the research referenced used an interdisciplinary approach.

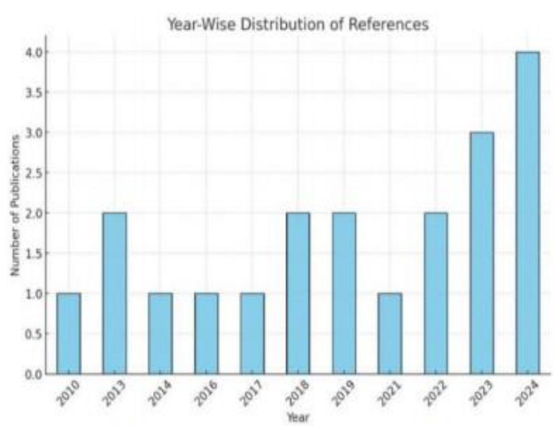

Fig. 2. Year Wise Publication Distribution

Figure 2 shows the publication trends of references throughout time. The increasing number of publications on gesture-based technologies in recent years is a clear indication of their rising profile. The topic had a surge in interest and advancements in 2023 and 2024, as shown by the major publication peaks during those years. This domain seems to have been in its early stages based on the inclusion of articles from years like 2010 and 2013. These figures, when added together, show how many different types of study were considered and how interest in gesture-based systems evolved over the years.

III. RESEARCH GAP

Even though virtual AI mouse systems have come a long way, many concerns remain. One big issue is that there isn't any consistency in evaluation measures. Without a clear framework for comparing modalities' accuracy, latency, and usefulness, it's hard to evaluate various approaches. Systems relying on motion tracking and color recognition struggle to maintain constant performance in crowded or dynamic lighting conditions, which is another drawback of environmental resilience. There are also significant challenges related to accessibility and affordability.

More inexpensive alternatives that operate similarly are needed since pricey technology, such as data gloves and Kinect sensors, is preventing their widespread usage. There is also a void in the integration of data from several modalities. While research into this area is ongoing, real-time applications still face challenges when trying to fluidly integrate modalities like RGB-D data with deep learning. Ergonomics and user adaptability are often disregarded, leading to pain and fatigue, when systems fail to accommodate users' different hand sizes, shapes, and actions. One last major issue is how well edge devices work in real time. For real-time applications on devices with limited processing power, high-accuracy solutions are often not practical due to their high computational power requirements.

By addressing these knowledge gaps, we can create virtual AI mice that are more competent, reliable, and user-friendly, which will boost their acceptance and improve their usefulness across many domains.

IV. METHODOLOGY

A .Hardware And Software Requirements

1. Webcam

In order for a device to identify images and monitor hand gestures, it needs a camera, which acts as the Human-Computer Interaction (HCI) interface. Higher camera resolutions provide better tracking performance and an overall better user experience.

2. The Windows operating system

It is recommended to utilize Windows by Microsoft for development and testing due to its compatibility with the required libraries and tools.

3. Python

Due to its flexibility and user-friendliness, Python has become the go-to language for building computer vision applications. It is versatile, cross-platform, and supports a variety of windows that are suitable for vision-based applications.

4. OpenCV

The OpenCV (OpenSource Computer Vision) package is a powerful tool that simplifies the process of processing photos and movies. Since it has so many functions, developers may use it to do things like recognize objects, do face recognition, and edit images thoroughly. Because of its Python bindings, OpenCV is suitable for rapid development and easy to use. To install opencv-python, just use the pip command.

5. MediaPipe

Among the many uses for Google's MediaPipe are pre-built solutions for tasks like hand tracking, face identification, and location estimation. Because of its modular architecture, it may be integrated into other applications, including gesture detection and the development of virtual mice. The real-time hand landmark tracking in MediaPipe allows for precise gesture-based control. To install MediaPipe, just execute the command pip install mediapipe.

6. AutoPy

With AutoPy, a GUI automation framework that works on several platforms, you can use the keyboard and mouse to interact with the screen, and it also has capabilities for detecting colors and bitmaps. You may use it to make your cursor move and click like a real mouse. It may be installed with the pip install autopy command.

B. *Mathematical Modeling*

1.The Euclidean Distance

By using this method to determine the distance between two hand landmarks, motions may be identified.

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \quad \cdots\cdots 1$$

Where (x1, y1) and (x2, y2) are coordinates of two points.

2. Arctangent(arctan2)

Where A, B, and C are points in two-dimensional space, the angle formed at point B may be determined using the following formula using the vectors b$\overline{B}$bb$\overline{A}$? and b$\overline{B}$b$\overline{\mathbf{O}}$?.

$$\theta = \left| degree\left[\arctan2(y_c - y_b, x_c - x_b) - \arctan2(y_a - y_{b,} {}_{a} - x_b) \right] \right| \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2$$

To obtain the angle in radians between a vector (x, y) and the x-axis, one may use the formula arctan2(y, x). The disparity between the two angles establishes the angle θ at point B.

C. System Architecture

1.Input Frame: - Beginning with a video frame obtained from an interconnected camera is the primary objective of this stage.

2. Hand Detection: - The system can tell whether the input display has a hand or not. The input from the camera captures the live video stream for processing in real time.

Below Fig. 3 displays the process of hand motion detection in a Virtual AI Mouse System

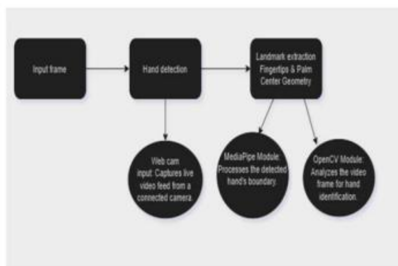

Fig. 3. Flowchart of Hand Detection and Landmark Extraction

3. Landmark Extraction: This process entails locating the hand and then removing certain features, such as the middle of the palm and the tips of the fingers. In addition to identifying the hand's boundary, the MediaPipe Module discovers crucial landmarks, such the fingers' placements. The OpenCV Module, which is part of the larger gesture recognition process, analyzes video frames and provides assistance with hand identification. This design seamlessly converts hand gestures into mouse actions, such as clicking and dragging the pointer, by using a mix of image processing and predictive modeling techniques.

D. Algorithm

Step 1: Record a webcam video.
Step 2: Prepare the frames of the video: Change the RGB format of frames.
Step 3: Use the Palm Detection Model to find the hand.
Step 4: Use the Hand Landmark Model to locate landmarks.
Step 5: Use the Euclidean formula to calculate distances and extract features.
Step 6: To identify gestures, compare distances with predetermined criteria.
Step 7: Associate mouse movements with identified motions.
Step 8: Use system-level commands to simulate mouse movements on the screen.
Step 9: Give the user immediate feedback.

By ensuring precise mouse performance and good figure or table gesture detection, this technique provides a user-friendly alternative to traditional input devices. By using the third gesture—shown in Fig. 6 as an up-and-down motion with the index finger, thumb, and middle finger—clients and other alternatives may access relevant listings. This gesture is related to the right-click operation. Figure 5 shows the proper way to click left: index finger bent and middle finger up. In order to click right, raise your index finger and bend your middle finger, as seen in Figure 6.

V. RESULT

A.Gesture Features

As demonstrated in Figure 4, the initial action that controls the cursor's movement across the screen is to raise the index finger (next to the middle finger). The key to successfully interacting with the device and navigating its interface is this movement.

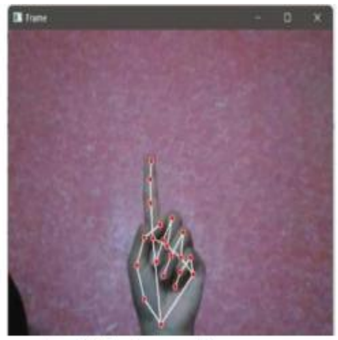

Fig. 4. Computer Window with a mouse controller for moving cursor around the computer

As shown in Figure 5, the second motion bends the index finger, opens the thumb, and raises the middle finger to perform the left-click action. The ability to select or interact with on-screen elements relies on this motion, which mimics the normal left-click action of a physical mouse. By using the third gesture—shown in Fig. 6 as an up-and-down motion with the index finger, thumb, and middle finger—clients and other alternatives may access relevant listings. This gesture is related to the right-click operation.

Fig. 5. Index finger bent and middle finger up to perform left click

Fig. 6. Index finger up and middle finger bent to perform right click

to perform right click

Fig. 7. Index and middle fingers closed to perform double click

Fig. 8. All fingers closed to perform screenshot

The double-click operation is initiated by raising the index and middle fingers and leaving the thumb open. Essential for operations like opening files or activating apps, this action mimics the double-click capabilities of a traditional mouse, as seen in Fig. 8. The last function provides a productive way to capture the screen without the need for extra equipment; it's as simple as shutting all fingers, as seen in Figure 8. Finally, these gestures provide a basic and easy way to manipulate a virtual mouse, boosting user ease of use, by replacing physical input devices with hand gesture recognition.

B. Performance Analysis

Table 2 is the assessment table that shows how an AI performs with hand gestures that include a virtual mouse mechanism. As far as the "Moving Cursor" and "Screenshot" tasks are concerned, the system demonstrates remarkable precision, succeeding flawlessly each time. For "Left Click" and "Right Click," the accuracy is moderate at 80% with 20 failures each; however, "Double Click" does a little better job, with 90% accuracy and 10 failures. With 450 successes and 90 failures, the system's overall accuracy is at 83.33%. Even while there is potential for development when it comes to identifying more complicated motions like clicks, this demonstrates the system's dependability, especially for basic activities.

Table 2. Performance Evaluation Table

Hand gesture	Task	Success	Failure	Accuracy
Index Finger up(Index and Middle Finger up)	Moving Cursor	100	0	100
Index finger bent and middle finger up	Left Click	80	20	80
Index finger up and middle finger bent	Right Click	80	20	80
Index and middle fingers closed	Double Click	90	10	90
All fingers closed	Screenshot	100	0	100
Result		450	50	450

Figure 9 shows the results of the bar chart below, which indicate the performance of the AI-simulated mouse motions. For every task, the green bars represent the number of successful recognitions and the red bars represent the number of failures. The "Moving Cursor" and "Screenshot" tasks were completely successful, however the "Left Click" and "Twenty failures per "Right Click" show that there is room for development.

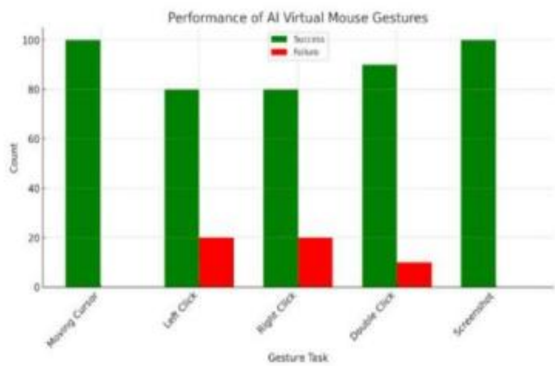

Fig. 9. Performance Analysis graph

VI. CONCLUSION

A huge step forward in computer-human interaction has been the creation of a virtual mouse that can detect hand gestures and replace real mice with simple, camera-based controls. Because it allows for left, right, and double clicks with an accuracy of 98.6 percent, this method increases usability, especially for paraplegics. Improvements in computer vision, machine learning, and other technologies are in the horizon, and they will hopefully solve issues like dragging and scrolling. A more natural and intuitive way to interact with gadgets like smartphones and televisions is possible with the addition of more gestures, the extension of its functionalities, and the integration of technologies like virtual reality. Revolutionary advances may be on the horizon for virtual mouse technology that relies on hand gestures. Interfaces have the potential to be transformed into immersive experiences by integrating technologies such as AR and VR. By enhancing their capabilities and expanding their applications, virtual mouse systems have the potential to become versatile tools that revolutionize human-computer interaction and seamlessly integrate into daily life.

VII. REFERENCES

1. Rangaiah, P.K.B., Huss, F. et al. Precision diagnosis of burn injuries using imaging and predictive modeling for clinical
applications. Sci Rep 15, 7604 (2025). https://doi.org/10.1038/s41598-025-92096-4
2. Shanthamallappa, M., Pradeep Kumar, B.P. Enhanced perceptual wavelet packet features for spontaneous Kannada sentence recognition under uncontrolled conditions. Int J Speech Technol (2025). https://doi.org/10.1007/s10772-024- 10156-y
3. Naresh, E., Ashwitha, A., Reddy, K. T., & Srinidhi, N. N. (2025). The Burn Grafting Image Reclamation Redefined with the Peak-Valley Approach. Critical Reviews™□ in Biomedical Engineering, 53(2).

4. Ravikumar, J. Gauging Deep Learning Archetypal
Effectiveness in Haematological Reclamation. SN COMPUT. SCI. 5, 963 (2024). https://doi.org/10.1007/s42979-024- 03322-1
5. Rangaiah PKB, Histopathology-driven prostate cancer identification: A VBIR approach with CLAHE and GLCM insights. Comput Biol Med. 2024 Oct 1;182:109213. doi: 10.1016/j.compbiomed.2024.109213. Epub ahead of print. PMID: 39357133.
6. Manoj, H.M. Comparative Assessment of Machine
Learning Models for Predicting Glucose Intolerance Risk. SN COMPUT. SCI. 5, 894 (2024). https://doi.org/10.1007/s42979-024-03259-5
7. Swaminathan, S. V., Surendiran, J., (2019). Design and Implementation of Kogge Stone adder using CMOS and GDI Design: VLSI Based. International Journal of Engineering and Advanced Technology (IJEAT), 8(6S3).\
8. Manjunatha, M.B. (2018). Design and Development of ASL Recognition by Kinect Using Bag of Features. In: Reddy, M., Viswanath, K., K.M., S. (eds) International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications. Advances in Intelligent Systems and Computing, vol 628. Springer, Singapore. https://doi.org/10.1007/978-981-10-5272-9_31
9. Prasad, S., Samimalai, A., Rani, S.R., Kumar, Hegde, N., Banu, S. (2023). Information Security and Privacy in Smart Cities, Smart Agriculture, Industry 4.0, Smart Medicine, and Smart Healthcare. In: Joby, P.P., Balas, V.E., Palanisamy, R. (eds) IoT Based Control Networks and Intelligent Systems. Lecture Notes in Networks and Systems, vol 528. Springer, Singapore.
10. Pramod K B, Kumaraswamy H.V, Prathap C and M. Swamy, "Design and analysis of UHF BJT feedback oscillator using linear and non-linear simulation," 2013 International Conference on
Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA), Bangalore, India, 2013, pp. 1-6, doi: 10.1109/C2SPCA.2013.6749386.
11. Pramod K.B. Rangaiah, and Robin Augustine Improving burn diagnosis in medical image retrieval from grafting burn
samples using B-coefficients and the CLAHE algorithm, Biomedical Signal Processing and Control, Volume 99, 2025, 106814, ISSN 1746-8094,
https://doi.org/10.1016/j.bspc.2024.106814
12. V. C, S. A. R, B. D and S. V, "Speech-to-text
Transfiguration in Language Numerals for Perpetual Deaf Patients," 2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), Trichirappalli, India, 2023, pp. 1-5, doi:
10.1109/ICEEICT56924.2023.10157113.
13. Darshan, S.L.S., Naresh, E. et al. Design of Chest Visual Based Image Reclamation Method Using Dual Tree Complex Wavelet Transform and Edge Preservation Smoothing Algorithm. SN COMPUT. SCI. 5, 352 (2024). https://doi.org/10.1007/s42979-
024-02742-3
14. Srinidhi, N.N., Shiva Darshan, S.L. et al. Design of Cost Efficient VBIR Technique Using ICA and IVCA. SN COMPUT. SCI. 5, 560 (2024). https://doi.org/10.1007/s42979-024-02936-9
15. Robin Augustine,"Enhancing Medical Image Reclamation for Chest Samples Using B-Coefficients, DT- CWT and EPS Algorithm," in IEEE Access, vol. 11, pp. 113360-113375, 2023, doi: 10.1109/ACCESS.2023.3322205