

An Efficient Vision-Based Algorithm for Hand Pose Estimation and Cursor Control

Dinesh Swami^{1,*}, Somineni Rohith Chowdary², Thumati karthik³, R. Hanish Kumar⁴, N. Dharmendra⁵

Abstract

Conventional input devices like the keyboard and mouse are no longer necessary because gestures are becoming more popular as the most natural way to interact with computers. This project has demonstrated a novel real-time system that lets users control their computer cursors using simple hand gestures. The system facilitates an easy operation for us in terms of the use of a cursor by using techniques and hand mark detections. The software maps the coordinate points of the detected finger tips according to their positions by using mapping techniques. Smooth, jitter-free cursor movement. Increasing precision along with user comfort and confidence while using the system. To the overall, this is a light solutions solution offering a feasible and effective alternative to the standard pointing devices with no necessary investment in the costly or specialized hardware. It is a system that picks and reacts to simple movement of hands to the computers (desktops and the laptops) on which the standard webcams are fitted. It also allows one to perform with ease the activities of the cursor-moving, clicking, double-clicking and dragging and scrolling-that are a part of the daily work on the computer. It can be controlled by the fact that it does not require the touch of any object, hence fulfilling the needs of people who cannot move freely in several places like hospitals and educational institutions, this technology can also be applied to understand the gesture computer. Interaction The system can also be user-specific by the system to be built in a modular way; thus, the piping will work under different settings and under different lighting conditions. This system uses a coordinate normalization, dynamic thresholding and preprocessing to attain constant results in the detecting various backgrounds and a large assortment of camera relative positions. The time filtering will be inserted in the decision logic since cursor action, like clicking and dragging, could only occur when the voluntary gesture of the user occurs. This lowers the fatigue or weariness and therefore the interface is reliable to be used in the long-term. The reality also evades the redundant processing overheads and provide fewer steps to compute with the aim of dwelling down on real time performance. Because all the modules of the pipeline facilitate the ease of experience (i. e., frame acquisition), landmark identification, gesture classification and cursor mapping, the system may respond to the input, and it may also depict the fact that the hand control via the camera is not a science-fiction fiction. Furthermore, it is also scalable, and there is no need to find large-scale size of the architecture complicated since it is impossible to add more gestures or multi-hand messages.

Gesture interfaces are also coming into existence, and a further developed interface architecture can be developed on the system: virtual drawing and 3D manipulation or implementation in terms of an augmented (or virtual) reality architecture. In addition to the general applications of the technology, there would be other specific applications, such as touchless public kiosks, interactive learning devices or access devices that would be used by individuals with motor impairment. Overall, the nascent science of the natural-user interfaces will be advanced with the assistance of the project, through a convenient, precise, and economical gesture-based cursor system. It shows that with careful algorithms and

*Author for Correspondence

Dinesh Swami
E-mail: dkswami@gmail.com

¹⁻⁵Department of computer science & Engineering Parul Institute of Engineering & Technology Vadodara, Gujarat, India

Received Date: February 09, 2026

Accepted Date: February 26, 2026

Published Date: March 10, 2026

Citation: Dinesh Swami, Somineni Rohith Chowdary, Thumati Karthik, R. Hanish Kumar, N. Dharmendra. An Efficient Vision-Based Algorithm for Hand Pose Estimation and Cursor Control. Research & Reviews: Discrete Mathematical Structures. 2026; 13(1): 24–35p.

heavy-load processing applications, a motion of the hands can be projected down to the acceptable and understandable computer directions and this will allow a far more interactive and involving on-screen interaction experiences.

Keywords: Gesture-based interaction, hand gesture recognition, computer vision, real-time cursor control, human-computer interaction.

INTRODUCTION

An efficient vision-based algorithm for hand pose estimation and cursor control focuses on enabling natural human-computer interaction without physical devices like a mouse. The system typically uses a standard camera to capture real-time video input, from which hand features are detected and tracked. Advanced computer vision techniques such as background subtraction, skin color segmentation, and contour detection are used to isolate the hand region. More robust approaches incorporate deep learning models, such as convolutional neural networks (CNNs), to accurately identify key hand landmarks and gestures under varying lighting and background conditions.

Once the hand pose is estimated, specific gestures are mapped to cursor actions. For example, moving the hand controls cursor position, while gestures like pinching or tapping simulate mouse clicks. To improve efficiency and responsiveness, algorithms often include noise filtering, motion smoothing, and predictive tracking techniques such as Kalman filters.

The system aims to achieve low latency, high accuracy, and robustness across diverse environments. Applications extend beyond basic cursor control to areas like virtual reality, assistive technologies, gaming, and touchless interfaces. Overall, such algorithms contribute significantly to intuitive, contactless interaction systems in modern computing environments.

LITERATURE REVIEW

Gesture based human computer interaction has evolved, as simple webcam-based detection has been developed up to some elaborate deep-learning systems. In order to detect simple gestures, early studies such as Kumari. [1-22], Grif and Farcas [25, 26] and Dehankar. Color segmentation, centroid detection and morphological processing were used in [4]. The methods were very delicate to their changes in lighting and background noises, despite allowing control of the cursor in a basic manner. Ojha and Rajan later on [12] and Cao [24] came up with a recognition method that employed SVM and histograms that were more efficient in computation but could not stand up in complex scenarios. Increasing desire of the contactless interface led to more useful virtual mouse technology developed by researchers.

Baheti. [5], Kanaka Raju. [6], Apoorva. [7] and Sahiwala. [15] implemented gesture-driven control of the cursor, clicking, sketching, and navigation of files. These works showed how hand signs can be employed in real-life tasks with computers. Although Kumar et al. [9] and Godbole et al. [10] extended gesture control to presentations and voice-integrated system, Soroni. Gesture-based virtual blackboards were introduced by [8]. All in all, the literature considered permanent issues, including sensitivity of illumination, occlusion issues, intermittent clicks, and absence of user assessment in practical conditions.

Despite these disadvantages, the findings record a vivid observation on trustworthy, camera-based, real-time. Zholshiyeva. Deep learning frameworks employed by [23] in order to make the implementation more accurate require adaptation to different types of users. Jayalakshmi brought about multi-systems that used hand gestures and eye tracking. [13] and Surya. [19]. These enhance the quality of interaction, as well as reducing false detections.

Emporio was an article [1], that attracted attention to the continued gesture recognition, which gave benchmarks and as the research continued, more lax and more adaptive and practice-driven virtual mouse technologies were developed. Kanaka Raju. [5], Baheti. Yu. [3] developed categories of neural models that were normalized on Raspberry Pi and binary roles running in real time. Emporio was on a continual trend of gesture recognition. It was just benchmarks that were offered, and as additional application based virtual mouse system were emerging, which caused its rise. Baheti. [5], Kanaka Raju. [6], Apoorva. [7] and Das et al. [15] applied classical hand-gesture pipelines to system level processes drawing, file processing and multi-purpose control of

a pointer.

These systems formed the reality of the fact that the interfaces that use a gesture recognition tool through a webcam can be used as the alternatives to the physical devices when performing the usual processing tasks of the regular computing at a reasonable level of accuracy with some difficulties Zholshiyeva. This further was enhanced by [23] in relation to the deep-learning frameworks which had the capacity to adapt to a number of users. Jayalakshmi. [13] and Surya. The existing systems as presented by [19] included the hand gestures together with eye tracking. It will be possible to avoid misleading detecting and improve the quality of communication with the help of such solutions. There were some similar works which were domain specific HCI applications. Soroni. Gosse was also developing a gesture-based virtual blackboard (Kumar [8]).

The 9 gestures were used to control presentations, virtual drawing and volume Godbole control. Among the experiments that [10] carried out was to examine the interaction between hand gesture and voice controls to ascertain that lesser reliance on hardware can be employed, however, Reddy. More immersive control was achieved in [16] using the hybrid hand-eye gesture mechanisms. This brings to the conclusion that generalization of cross-educational, cross-productivity, and cross-accessibility may be provided through gesture interfaces (Table 1).

Table 1. Literature survey

Authors	Title	Year	Findings	Gaps
M Emporio, Amirpouya Ghasemaghaci, Joseph J. Laviola Jr., and Andrea Giachetti	Continuous Hand Gesture Recognition: Benchmarks and Methods	2025	Examine extant standards of incessant gesture recognition, and discover them to be limited in quality and quantity, using up non-standard approaches. The vast majority of studies are aimed at categorizing pre-segmented gestures and instead of identifying them in a continuous stream.	New bench- marks that capture realistic conditions of the world and standardized measures of evaluation are required to deal with the high false positives and misses of renderings of gestures in existing systems.
Issam El Magrouni, Abdelaziz Et-taoufik, Siham Aouad, and Abderrahim Maizate	Hand Gesture Recognition for Virtual Mouse Control	2025	Created a Python-implemented virtual mouse utilizing finger angles and hand silhouette-to-convex hull ratio having 93.83% accuracy and 34 ms response time.	There are still difficulties keeping the accuracy since it is subject to varying lighting conditions, shifting hand orientations, and can be used by different users. The next generation of work needs to concern more flexible algorithms.
Alyssa Yu, Cheng Qian, and Yifan Guo	A Real-Time Hand Gesture Recognition System on Raspberry Pi: A Deep Learning- Based Approach	2024	Developed a real time gesture recognition system on inexpensive boards such as Raspberry Pi with more than 96% accuracy, with a lean CNN and model quantization to size reduction to 75%.	The system shifts model training to a server; a significant borne-down asset is that there is no on-device training and updating functions.
A.V. De- hankar, V. M. Thakare, and Sanjeev Jain	Detecting Centroid for Hand Gesture Recognition using Morphological Computations	2017	On the detection of the centroid of a static hand gesture in an image on a uniform background, it is proposed that a morphological operation-based method can work in MATLAB. a pre-processing step.	Only cen- troid detection is discussed in the research. Similar to the use of actual gesture recognition, future research is required in implementing thinning and end- point identification.
Akshat Baheti, Dev Patwa, and Sachin Gajjar	Gesture Based Com-puter Control System	2023	Shows an AI virtual mouse at a webcam and using one of the Medi- aPipe to locate 21 3D hand landmarks to control the volume and cursor in real-time using the mouse lms identification time.	It is possible to also improve the system to identify both hands and to manipulate keyboard capabilities.
B. Siva Kanaka Raju, Ch.Shiva Prasad, A.Pushpita, and K.Karthik	Gesture-based File Management and Painting Application using Human Com- puter Interaction	2025	Created a gesture applica- tion to handle files and paint with MediaPipe and RNNs, and obtained great results in terms of accuracy (88-95%) and smoothing (smooth performs) at ~30 fps.	The system experiences problems with mis- classification of complex / overlap- ping movements and ambiguity with multi-user hence the need multi-user tracking.
Mariam Fathima A, Mounica V, and M. K. Kirubakaran	Revolutionary Cursor Control Systems En-hancing Human Com-puter Interaction	2024	Introduced a virtual mouse system on AI and uses hand gesture and speech recognition to control such functions as brightness, volume, web search and others,including a focus on inclusive to users with disabilities.	Fills the gap of adapt- ability and inclusivity of the exist- ing cursor control systems that lack suitability in a variety of areas such as design, gaming, or accessibility.
M. M. Sahi- wala, D. D. Jakasaniya, S. R. Singh, K. M. Patel, and Prof. Deepthi Pawar	Virtual Mouse using Coordinate Mapping	2023	Introduced a computer-generated mouse simulator, which uses MediaPipe and OpenCV to map finger position to cursor movements, and attempt to provide a feel similar to a hardware-based feel, without colored markers.	There are still problems with the accurate monitoring and analysis of hand movements. The sys-tem is designed to prevent use of colored gloves which are necessitated by other systems.

Authors	Title	Year	Findings	Gaps
A. Apoorva, A. Shankar, A. Bondi, A. R. Rani, G. Jaiswal, A. Sharma	Gesture-Controlled Mouse and Finger Air Writing	2024	Integrates with virtual mouse gestures with free-air handwriting recognition, allowing the user to draw or write with tracked fingertip movement.	Accuracy Air-writing is often volatile, and light conditions have to be steady, and accuracy deteriorates with prolonged sessions.
F. Soroni, S. Al Sajid, M. N. H. Bhuiyan, J. Iqbal, M. M. Khan	Hand Gesture Based Blackboard Using Webcam	2021	Introduces a virtual blackboard, which is used on a webcam and enables a user to sketch or write something based on identified hand movements in real-time.	The range of drawing is minimal, the system cannot work with rotating hands greatly and the complexity of the background interferes with the work of tracking.
A. Kumar, H. Anandaram, R. Pandey, K. Joshi, K. Alam, S. Chaudhary	Hand Gesture based AI Controller for Presentation, Virtual Drawing and System Volume Management	2024	Designs a flexible gesture recognition controller based on AI which combines presentation navigation, digital drawing, and volume control of the system.	The similarity of gestures incurs command ambiguity, and poor performance both in crowded scenes and with variable hand movements by the user.
A. Godbole, V. Gondke, K. K. Devadkar	Hand Gesture Identification and Voice Command Based Hardware Reduction	2012	Adds voice command with gesture recognition to minimize the physical hardware use to control operations on many system features.	Noiselessness of voice recognition can be observed in noisy conditions, and the problem of gesture voice synchronization cannot be addressed in this case of using it by more than one person at a time.
P. Kumar S, S. Agarwal, D. Mustafa, D. Kumar, G. Ranjan	Hands-Free Mouse Control Using Facial Feature	2023	Takes facial landmarks to track the movement of the mouse without using the hands and provides the hand-free interaction aiming at the accessibility of many users.	Small-scale precision of the cursor, high interdependence on the quality of the camera, and possible privacy issues restrict applications.
D. Ojha, R. G. Rajan	Histogram Based Human Computer Interaction for Gesture Recognition	2019	Uses histogram-based color and gradient analysis to overcome and classify gestures effectively with low computation needs.	Very light to changes in illumination not robust to complicated gestures and worse than modern deep models.
M. Jayalakshmi, T. P. Saradhi, S. M. R. Azam, S. Fazil, S. D. S. Sriram	Multi-model Human-Computer Interaction System with Hand Gesture and Eye Gesture Control	2024	Uses a multimodal system of interacting with hand and eye gestures to have it more flexible and expressive through commands.	Useful injury there must be high calibration, a modal combination is difficult to implement, and the system suffers a high-frequency or irregular user motion.
N. L. Reddy, R. Murugeswari, Md. Imran, N. Subhash, N. V. K. Reddy, N. B. Adarsh	Virtual Mouse using Hand and Eye Gestures	2023	Combines hand gesture with movement and eye gesture with clicking, which increases usability by more users with reduced mobility.	There is a decrease in the accuracy of eye-tracking with webcams, multimodal conflicts observed and general system reliability is unreliable.
A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, J. S. Nayak	Hand Gesture Recognition for Human Computer Interaction	2017	Small gesture recognition implementation Early gesture recognition using segmentation and simple shape features to provide simple HCI interactions.	In dynamic environments, recognition is poor, and is not robust to clutter as well as capable of addressing complex gestures.
H.-S. Grif, T. Turca	Human hand gesture based system for mouse cursor control	2018	Adopts an elementary gesture-based cursor system on the basis of contour extraction and simple hand movement tracking algorithms.	The movement of the cursor is jerky, there is a lowering of performance when subjected to changing light, and the gesture vocabulary is low and rigid.
S. Surya P, S. Santhosh, S. Kumar P, M. Deepa	Cursor Movement by Hand Gesture	2024	Converts real time hand movement to cursor movement providing hands-free movement with simple vision algorithms.	Lacks accuracy in fine control, challenging in tracking regularly, and does not have any advanced filtering to stabilize movement.

Authors	Title	Year	Findings	Gaps
M. N. Das, J. Tripathy, E. R. Kumar, A. Pottabathula, Y. R. Bandi, M. S. Surepalli, P. Dhoragolla	Hands-Free Mouse With Gesture And Voice Control	2025	Integrates gesture and speech recognition to form an entirely hands free mouse that meets the requirements of accessibility and overall interaction.	Voice recognition suffers in noisy conditions, the lag in the system varies and the coordination of modalities is not optimized completely.
N. Beisov, G. Madyarova, N. Kerimbayev	Gesture recognition technology: a new dimension in human-computer interaction interface	2024	Surveys The emerging gesture recognitions technologies and present trends, uses and futur problems in HCI systems	Offer no experimental comparison, and not quantitative analysis and scanty technical depth of which can be practically considered.
P. Kumari, S. Singh, V. K. Pasi	Cursor Control using Hand Gestures	2016	Design early gesture recognition through simp segmentation mechanisms over webcams in order to manipulate simple cursor behaviors.	Sensible to lights and foreground variations, no sophisticated gesture recognition and cannot follow trustworthy click behaviour.
L.Zholshiyeva, Z. Manbetova, D. Kaibassova, A. Kassymova, Z. Tashenova, S.Baizhumano v.A.Yerzhanov a, Kaikhyrbay	Human-machine interactions based on gesture recognition using deep learning methods	2024	Trains deep learning models to achieve better gesture-recognition speed accuracy, and resilience in a variety of gestures and user scenarios.	There is a disadvantage of heavy model size, dataset diversity, and syste needs the computing power of a GPU.
W.Cao	Application of the Support Vector Machine Algorithm based Gesture Recognition in Human-computer Interaction	2019	SVM Handcrafted features to classify gestures with reasonable accuracy to small-scale gesture sets only.	SVM does not scale well with large sets of gesture vocabularies, needs large amounts of feature engineering, and is not as effective as deep

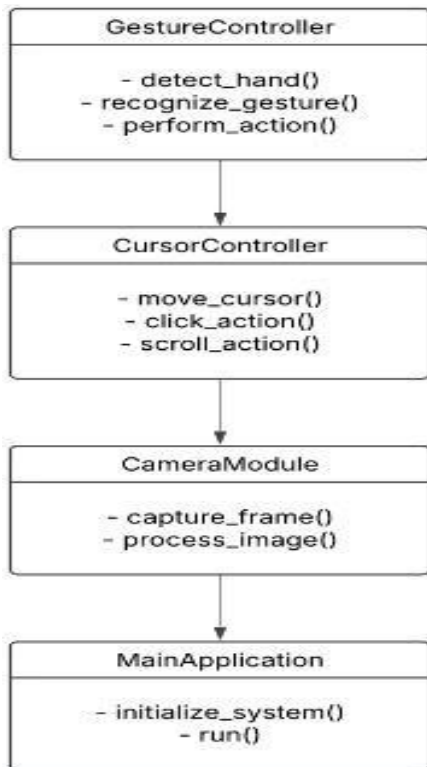


Figure 1. The proposed system follows a structured computer- vision pipeline.

METHODOLOGY

The proposed system follows a structured computer- vision pipeline designed to translate real-time hand gestures into mouse-cursor actions. The methodology consists of five major stages: video acquisition, hand detection, landmark extraction, gesture interpretation, and cursor-control execution. Each stage is optimized for accuracy, responsiveness, and smooth user interaction. Figure 1 shows the working of Gesture controller, Cursor Controller, Camera Module and Main Application.

Distance Formula (for Click Gesture Detection)

Used to check whether the thumb and index finger are close enough to trigger a click.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Where:

(x_1, y_1) → Thumb fingertip coordinates

(x_2, y_2) → Index fingertip coordinates

Coordinate Mapping Formula (Camera → Screen Mapping)

Used to map your webcam's frame to your monitor resolution.

$$X = \frac{x_{cam}}{W_{cam}} \times W_{screen} \quad (2)$$

$$Y = \frac{y_{cam}}{H_{cam}} \times H_{screen} \quad (3)$$

Where

x_{cam}, y_{cam} → Fingerprint position in camera frame

W_{cam}, H_{cam} → webcam frame width/height

W_{screen}, H_{screen} → monitor width/height

Smoothing / Low-Pass Filter Formula

Used to reduce jitter and shaky cursor movements.

$$P_{smooth} = \alpha P_{new} + (1 - \alpha)P_{old} \quad (4)$$

Where:

P_{new} → new fingertip position

P_{old} → previous cursor position

α → smoothing factor (0.1–0.3 recommended)

Video Frame Acquisition

A regular Webcam is employed and constantly captures live video frames with 30–60 FPS. All these frames are then rescaled and processed to ensure that the processing speed remains constant. The simple normalizations are done to reduce the influence of different lighting conditions and to enhance the appearance of hand features.

Hand Detection and Region Isolation

The system uses a powerful image-processing of the hand of the user with each frame. The frames are converted into matching colour space and thresholding or landmark-based detection is performed in order to extract the portion of the hand. The image of the area can be enhanced by using noise processing (Gaussian blurring, morphological operations, and so on) to remove false alarms.

Landmark extraction and fingertip identification

It is aware of such landmarks as finger points, finger joints and palm points. The first point of focus which happens to be the index fingertip which is the most important tool of pointing. The geometric relationship of the landmarks are also calculated with an intention of determining the location of the fingers, presence or relative position to determine a gesture.

Gesture Recognition and Action Mapping

This system understands the user's gestures by analyzing the spatial position and movement of extracted landmarks.

- *Cursor movement*: the index finger tip coordinates are mapped onto the screen coordinate system.
- *Left click*: Triggered when the distance between the index finger and thumb falls below a predefined threshold.
- *Right click*: Using a specific combination of fingers or by making a closed fist gesture to open the context menu.
- *Drag/selection*: Detected when the clicking gesture is held while the index fingertip continues to move. The smoothing algorithm is applied to the fingertips coordinates to reduce

Coordinate Mapping and System Control

The coordinates obtained in the web camera feed are in pixel units, which is non-matching with the screen coordinate system. To this end, I used a normalisation and scaling routine whereby the position of gestures is proportionally mapped into the display of the system. The Python library of usual cursor movements and click operations is executed using the Python library of pyautogui with incredibly low latency.

Performance Optimization

It is optimised by removing unnecessary computations, storing the frames that can be reused, and focusing on lightweight models in the first place. I have smoothing filters and time validity checks and debouncing logic entered in to prevent accidental clicks or undesired jumps of the cursor.

Testing and Evaluation

I had tried the system on different background, lights as well as on different positions of the hands. This user test was conducted on the parameters of main indicators the most common being the accuracy, responsiveness and the frequency of the gesture recognition and the usability. I also prioritized the performance indicators like the number frame per second, stability of the mouse and the

successful clicks in order to ensure that there was stability in the system.

SYSTEM ARCHITECTURE DIAGRAM

Purpose

The project development shall then continue on to a design, development and trial of a viable real-time gesture-based control over the cursor to which will be capable of being intuitively controlled with the help of a low-end web-cam and with the help of the useful free-lightweight program. I can also demonstrate that instead of substituting and even augmenting what could be considered the classical pointers i.e. mouse and trackpad, with the presence of camera-based hand gestures, can be practiced on a daily basis, and even can be successful even without having any advanced and costly equipment mounted therein, which, in its turn, could not be attached to any generic computer.

HIGH-LEVEL OVERVIEW

Actors / External Entities

- User: performs hand gestures.
- Webcam delivered- Video frame supply.
- Operating System- new artificial events which are generated by the mouse.
- Other Existing Libraries used External libraries can be used: OpenCV, MediaPipe (in certain cases only), PyAutoGUI, NumPy.
- Manager. Strict control Tests which will be sent by the obtaining chain will be stored electronically in the test logs folder and the reading of the tests as industry records are highly controlled.
- System Boundary e.g. the processing of the video will be defaulted to be run on the machine of the user though that will not be shipped over the net or even the cloud.

PRINCIPAL COMPONENTS

Input Module (Camera Capture)

- *Implementation:* Bugging configure the web camera of the target frame rate and acquire the consecutive frames and store the same in processing queue.
- *Data:* This is the H W 3 value of a given array of RGB values (integers).

Preprocessing Module

- *Implementation:* resize, conversion colour space (BGR RGB or HSV), utilisation of histogram equalisation / CLAHE and blur will be executed keeping the aim of destroying noise and hence they will be given a rebuttal image, which is less noisy.
- *Data:* The pre-processed frame.

Detection & ROI Module

- *Identify hand part(s):* Output ROI box and reliability. May will either take classical methods (skin segmentation, contours) or landmark detector (MediaPipe).
- *Data:* ROI Mask and detection confidences.

Landmark Extraction Module

- *Anatomical data:* normalized ROI coordinates of finger 0-1.
- *Data:* Landmarks [id, x norm, y norm, visibility].

Gesture Interpretation Module

- *Responsibilities:* Applies geometrical principles and time as a help to classify gestures - point, click, double-click, drag, scroll. Rebounds and gesture state machine are in state Holds.
- Information Gesture Event = type, state, time, data.

Coordinate Mapping Module

- In regard to normalised landmark positions relative to the screen position (calibration and gain

was used). Feature and velocity limiter Smoothing (filter).

- *Data:* ScreenPoint = {X_px, Y_px}.

Smoothing & Filter Module

- *Parameters:* To perform the exponential smoothing / low-pass filtering of the cursor, a user can apply a parameter alpha; or an optional Kalman filter can be applied to perform advanced smoothing.
- *Formula:* $P_{smooth} = \alpha * P_{new} + (1-\alpha) * P_{old}$.

Control & Actuation Module

Gesture event classifier The new breakthrough in autonomous driving of using keyboard and mouse is simply an exposition to how such a like image recognition can be executed, PyAutoGui to execute Python code (and most importantly Lisp), to give a command to the operating system and impeccable factor based virtual assistants to place a call in providing directions to autonomous driving

UX & Feedback Module

Activities that will be implemented: Fetching of overlay preview window - cursor and ROI and gesture state. Use little visual acids of clicks. which serve in the calibration and testing.

System Management & Logging Module

Responsibilities: Activities that will be implemented: Fetching of overlay preview window - cursor and ROI and gesture state. Use little visual acids of clicks. which serve in the calibration and testing (Figure 2).

RESULTS AND ANALYSIS

We projected MediaPipe hand loci to OS actions like clicks and scroll, due to the strong monitoring-loop in real-time.

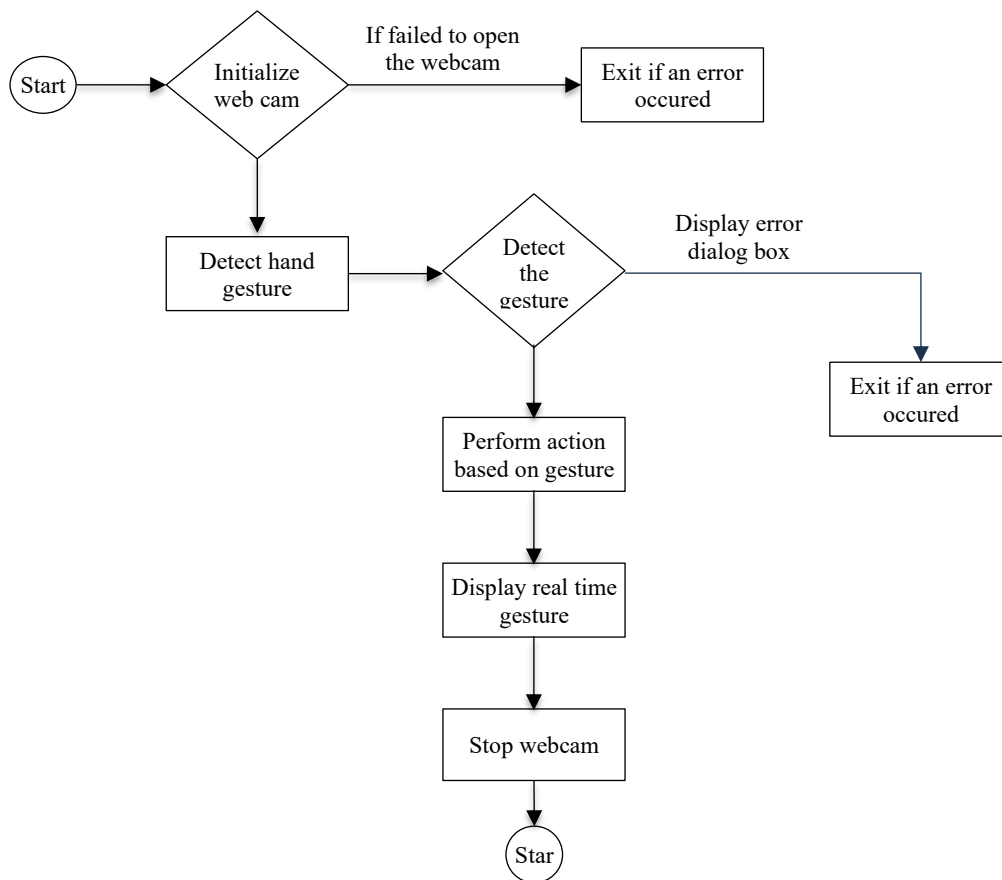


Figure 2. Work flow.

This performance evaluation shows that the cursor motion is the most desirable part of the primary navigation of a system with the accuracy up to 80-percent as presented in Figure 6. The lag of left and double-click features is at about 60 per cent. The disjunction of depth perceiving in pinch action of Figures 3–5, right-click and cursor-freezing are more stable at approximately 70 per cent. Irrespective of such insignificant differences, the system seems to be pretty serviceable as far as the use without contacts is concerned because it can recognize advanced finger patterns and give instant visual feedback (Table 2).

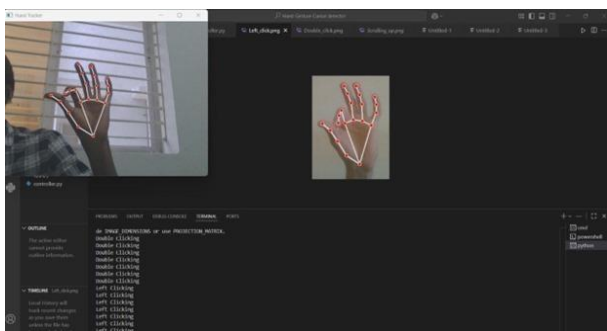


Figure 3. Double click gesture.

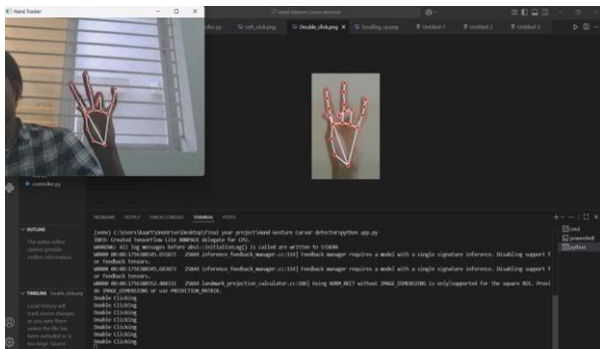


Figure 4. Left click gesture.

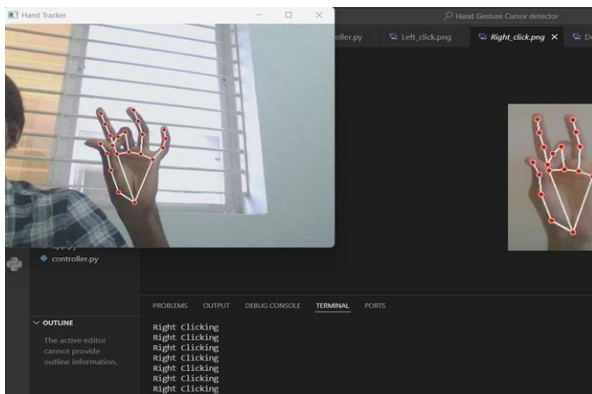


Figure 5. Right click gesture.

Table 2. Accuracy & observation.

Feature	Accuracy (%)	Observations
Cursor movement	80	Smooth and reliable across scenarios
Left click	60	Minor jitter observed occasionally
Right click	70	Some misclassification errors
Double click	60	Accuracy fluctuates under movement
Mouse freezing	70	Occasional freezing under rapid gestures

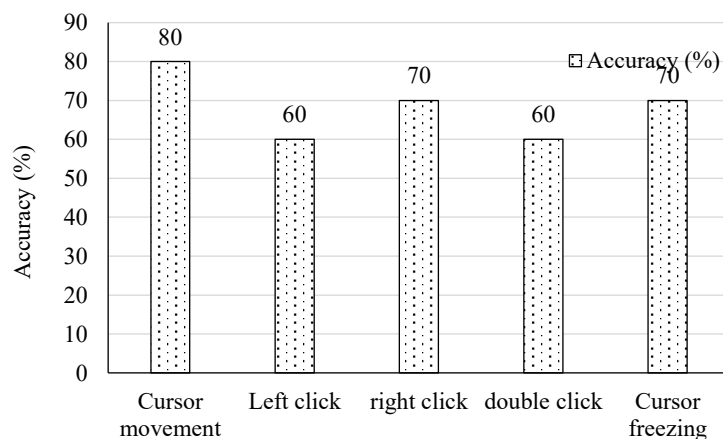


Figure 6. Gesture accuracy graph.

FUTURE SCOPE

The software can also include various advancements in the system like detecting more gestures like drag and drop and different clicks in variations. Tries to work very efficiently in different conditions

can also integrate with different technologies like AR/VR Can also work on integrating both voice and gesture to provide different ways to interact with the users.

The system will work on getting add on different gestures like drag and drop also different variations on clicking. using different technologies by integrating to this system will result in the different results.

CONCLUSION

This research defines that using of camera based hand gesture recognition will be a solution for interacting with the devices other than traditional way of interacting with the applications. The system is created to use simply and a knowledge of understanding complex gestures easily that works well in different conditions to whom have problems in operating the gestures. Future enhancements promise to broaden the gesture vocabulary and integrate multi-model commands for richer, more natural control experiences.

REFERENCES

1. M. Emporio, A. Ghasemaghaei, J. J. Laviola Jr., and A. Giachetti, Continuous Hand Gesture Recognition: Benchmarks and Methods, 2025.
2. J. El Magrouni, A. Ettaoufik, S. Aouad, and A. Maizate, Hand Gesture Recognition for Virtual Mouse Control, 2025.
3. A. Yu, C. Qian, and Y. Guo, A Real-Time Hand Gesture Recognition System on Raspberry Pi: A Deep Learning-Based Approach, 2024.
4. A. V. Dehankar, V. M. Thakare, and S. Jain, Detecting Centroid for Hand Gesture Recognition Using Morphological Computations, 2017.
5. A. Baheti, D. Patwa, and S. Gajjar, Gesture Based Computer Control System, 2023.
6. B. S. Kanaka Raju, A. Pushpita, C. S. Prasad, and K. Karthik, Gesture-based File Management and Painting Application Using HCI, 2025.
7. A. Apoorva, A. Shankar, A. Bondia, R. Rani, G. Jaiswal, and A. Sharma, Gesture-Controlled Virtual Mouse and Finger Air Writing, 2024.
8. F. Soroni, S. al Sajid, M. N. H. Bhuiyan, J. Iqbal, and M. M. Khan, Hand Gesture Based Virtual Blackboard Using Webcam, 2021.
9. A. Kumar, H. Anandaram, R. Pandey, K. Joshi, K. Alam, and S. Chaudhary, Hand Gesture based AI Controller for Presentation, Virtual Drawing and System Volume Management, 2024.
10. A. Godbole, V. Gondke, and K. K. Devadkar, Hand Gesture Identification and Voice Command Based Hardware Reduction, 2022.
11. P. Kumar S, S. Agarwal, D. Mustafa, D. Kumar, and G. Ranjan, Hands-Free Mouse Control Using Facial Feature, 2023. D.
12. Ojha and R. G. Rajan, Histogram Based Human Computer Interaction for Gesture Recognition, 2019.
13. M. Jayalakshmi, T. P. Saradhi, S. M. R. Azam, S. Fazil, and S. D. S. Sriram, Multi- model Human-Computer Interaction System with Hand Gesture and Eye Gesture Control, 2024.
14. M. Fathima A, M. V, and M. K. Kirubakaran, Revolutionary Cursor Control Systems Enhancing Human Computer Interaction, 2024.
15. M. M. Sahiwala, D. D. Jakasaniya, S. R. Singh, K. M. Patel, and D. Pawar, Virtual Mouse using Coordinate Mapping, 2023.
16. N. L. Reddy, R. Murugeswari, Md. Imran, N. Subhash, N. V. K. Reddy, and N. B. Adarsh, Virtual Mouse using Hand and Eye Gestures, 2023.
17. A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, and J. S. Nayak, Hand Gesture Recognition for Human Computer Interaction, 2017.
18. H.-S. Grif and T. Turca, Human hand gesture based system for mouse cursor control, 2018.
19. S. Surya P, S. Santhosh, S. Kumar P, and M. Deepa, Cursor Movement by Hand Gesture,

- 2024.
20. M. N. Das, J. Tripathy, E. R. Kumar, A. Pottabathula, Y. R. Bandi, M. S. Surepalli, and P. Dhoragolla, Hands-Free Mouse With Gesture And Voice Control, 2025.
 21. N. Beisov, G. Madyarova, and N. Kerimbayev, Gesture recognition technology: a new dimension in human-computer interaction interface, 2024.
 22. P. Kumari, S. Singh, and V. K. Pasi, Cursor Control using Hand Gestures, 2016.
 23. L. Zholshiyeva, Z. Manbetova, D. Kaibassova, A. Kassymova, Z. Tashenova, S. Baizhumanov, A. Yerzhanova, and K. Aikhynbay, Human-machine interactions based on hand gesture recognition using deep learning methods, 2024.
 24. W. Cao, Application of the Support Vector Machine Algorithm based Gesture Recognition in Human-computer Interaction, 2019.
 25. H.-S. Grif and C. C. Farcas, Mouse Cursor Control System Based on Hand Gesture, 2016.
 26. G. V. Bhole, S. Deshmukh, M. D. Gayakwad, and P. R. Devale, Implementation of Virtual Mouse Control System Using Hand Gestures for Web Service Discovery, 2024.