© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716554

Gesture Sense: Controlling Computer System Using Hand and Eye Gesture

RAJ PALANDE[1], ROHIT SURVE[2], ROSHAN PAWAR[3], KALPANA GANGWAR[4]
[1, 2, 3, 4] *Department of Computer, Engineering Mumbai University*

Abstract- Human–computer interaction has significantly improved with advancements in computer vision and AI. This paper presents an adaptive gesture-based control system for both physically challenged and general users. It operates in two modes: eye-controlled for users with limited mobility and hand gesture-based for standard users. Using a webcam, the system captures facial landmarks and hand movements to perform real-time actions like cursor control, keyboard input, app switching, brightness, and volume control. The proposed system provides an efficient, low-cost, and accessible alternative to traditional input devices.

Keywords—Gesture Recognition, Eye Tracking, Human-Computer Interaction

I. INTRODUCTION

Human–Computer Interaction (HCI) has advanced significantly with the rapid progress of computer vision and artificial intelligence [1]. Despite these developments, traditional input devices such as keyboards and mice still dominate computer interaction, though they are not always suitable for individuals with physical impairments [2]. This highlights the need for more inclusive and accessible alternatives. Gesture-based interaction systems have emerged as a promising solution by enabling users to control computer systems through natural actions such as hand and eye movements [3].

In recent years, hand gesture recognition has gained considerable attention due to its ability to support contactless interaction [4]. Using a camera and image processing techniques, hand movements can be captured, analyzed, and mapped to system commands such as cursor control, clicking, and application management, offering a more natural and intuitive user experience [5].

Eye gesture recognition is also an effective input method, especially for users with limited or no hand mobility [6]. Techniques such as eye tracking and blink detection allow the system to interpret eye movements and translate them into actionable commands, enabling independent computer use for physically challenged users [7].

The proposed system develops a unified gesture-based control framework using a standard webcam, integrating both hand gesture and eye movement recognition to perform multiple system-level operations [8]. It automatically switches to eye-based input when hand gestures are not feasible, ensuring continuous interaction. Eliminating specialized hardware such as sensor-based gloves improves affordability and simplifies deployment [9].

Overall, the system enhances accessibility and efficiency by providing an inclusive interaction mechanism for both disabled and non-disabled users, enabling more natural and effective computer interaction [10].

II. RELATED WORK

Gesture recognition systems have gained significant research attention due to their ability to enable natural and touchless human–computer interaction, particularly in assistive computing applications [3]. A wide range of approaches has been proposed in this field, each varying in terms of recognition accuracy, usability, and dependency on specialized hardware [4].

Gesture recognition systems are generally structured into two primary stages: data acquisition and gesture interpretation with classification [5]. In the data acquisition stage, user motion is captured using different sensing mechanisms. Early research commonly employed wearable devices such as data gloves and inertial sensors to accurately track hand and finger movements [6]. While these approaches provide high precision, they are often expensive,

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716554

intrusive, and unsuitable for continuous or everyday use [7].

To overcome these limitations, vision-based techniques using cameras have been extensively explored. Advanced systems utilizing depth sensors and stereo vision have improved recognition performance by capturing three-dimensional spatial information [8][9]. However, such systems require specialized hardware, which increases overall cost and limits large-scale adoption [10].

More recent research has shifted toward using standard RGB cameras combined with software-driven computer vision techniques for gesture recognition [11][12]. These approaches analyze real-time video streams without relying on additional hardware components. As a result, they offer a more affordable, portable, and practical solution for real-world applications [13].

In parallel, eye-tracking systems have been developed to assist individuals with severe motor impairments by interpreting eye movements, gaze direction, and blinking patterns for system interaction [14]. Traditional solutions often depend on infrared-based devices or specialized cameras, making them expensive and complex to deploy [15]. However, recent studies have demonstrated that reliable eye tracking can also be achieved using standard webcams combined with facial landmark detection techniques [16].

The gesture and eye recognition process typically involves multiple image processing stages, including noise reduction, segmentation, feature extraction, and classification [17]. Techniques such as Gaussian filtering and thresholding are used during preprocessing to enhance image quality, while morphological operations like erosion and dilation help refine detected regions [18]. The extracted features are then processed using rule-based logic or machine learning methods to map gestures and eye movements to specific system commands [19].

Despite notable progress in both gesture-based and eye-tracking technologies, most existing systems focus on a single interaction modality. Gesture-only systems are not suitable for users with severe physical impairments, while eye-tracking systems alone may not provide efficient interaction for general users due to slower response times and limited precision [20].

To address these challenges, the proposed system introduces a unified hybrid framework that integrates both hand gesture recognition and eye-based control. The system dynamically selects the appropriate interaction mode based on user capability and context. By utilizing a standard webcam and lightweight software processing, the approach eliminates the need for specialized hardware, ensuring low cost, high accessibility, and easy deployment for both disabled and non-disabled users.

III. PROPOSED GESTURE RECOGNITION SYSTEM

The overall system is divided into two main components: the back-end and the front-end. The back-end consists of three primary modules, namely the Camera Module, Detection Module, and Interface Module, as illustrated in Fig. 2.

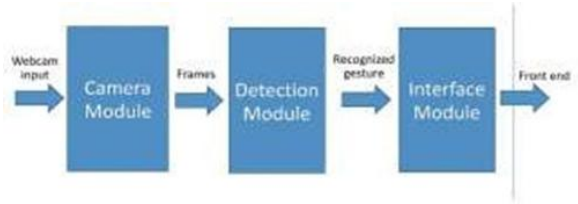

Fig. 2. Back-end Architecture

A) Camera Module

The Camera Module is responsible for capturing real-time visual input and forwarding it to the detection module in the form of video frames. In this system, a standard built-in webcam is used to acquire input, making the solution cost-effective and easily accessible. The module continuously captures frames from the video stream, “enabling detection of both static and dynamic gestures using only a standard webcam.

B) Detection Module

The Detection Module processes the frames received from the camera and identifies the gestures performed by the user. Initially, the captured images are converted into an appropriate color space to simplify processing. This is followed by pre-processing steps

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716554

such as noise reduction and smoothing to enhance image quality.

Next, segmentation and thresholding techniques are applied to isolate the region of interest, such as the hand or face, from the background. For hand gesture recognition, contour detection is used to extract hand shape and boundary information. By analyzing these contours and identifying finger positions, the system determines the corresponding gesture.

For eye gesture recognition, facial landmark detection is used to locate the eye region. The system then monitors eye movement and blinking patterns to interpret user commands. Once the gesture is recognized, the results are passed to the interface module for further action.

C) Interface Module

The Interface Module maps recognized gestures to system-level actions.and executes the corresponding operations. These actions may include cursor movement, clicking, application control, or other system-level tasks.

The front-end interface is designed with three display windows for better visualization and user feedback. The first window shows the live video feed along with the detected gesture label. The second window displays the detected contours, while the third window presents the thresholded image after processing.

Including contour and threshold views in the interface helps users understand how the system interprets input. It also allows them to adjust lighting conditions or camera positioning to minimize background noise and improve detection accuracy. This enhances the overall performance and usability of the system.

III. PROPOSED METHOD

We propose a marker less gesture recognition system, that follows a very efficient methodology as shown in fig. 3&4.

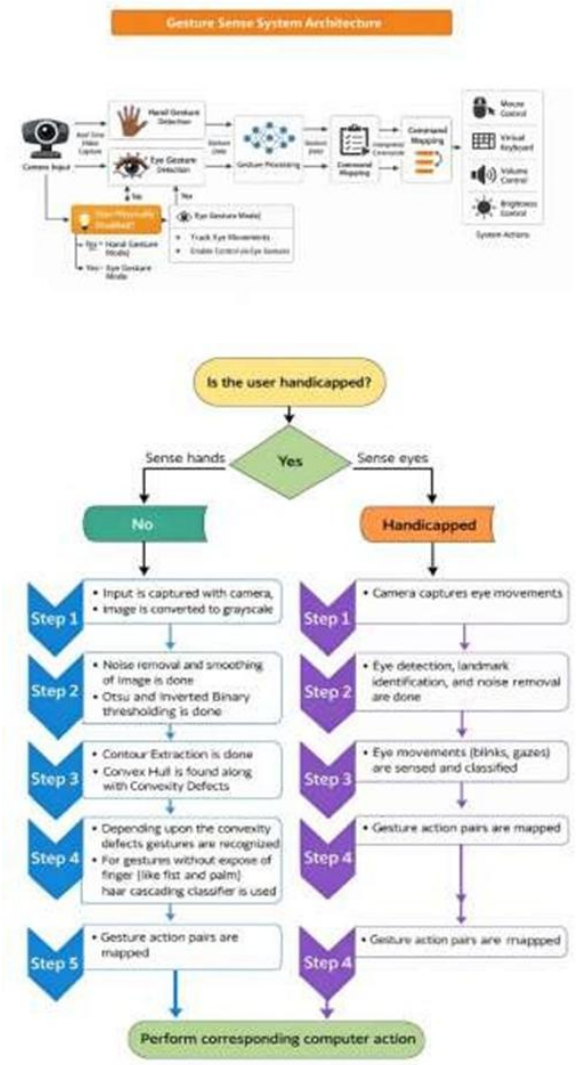

FIG. 3 PROPOSED METHOD FOR OUR GESTURE RECOGNITION SYSTEM

a) Noise removal and Image smoothening

The input image is initially captured in RGB color format and resized to a fixed dimension of 300 × 300 pixels to ensure consistent processing. It is then converted into a grayscale image, which reduces computational complexity and simplifies further image processing steps (as illustrated in Fig. 4).

Noise in an image refers to unwanted variations in brightness or color, typically introduced during image acquisition through the webcam. Such distortions can

negatively affect the accuracy of gesture detection and must be minimized.

To address this issue, a Gaussian filter is applied to smooth the image and reduce noise. Gaussian filtering works by convolving the image with a Gaussian kernel, where each pixel value is replaced by a weighted average of its neighboring pixels. This helps in suppressing noise while preserving important structural details of the image.

$$G(x,y) = Ae - \frac{(x-\mu x)2 + (y-\mu y)2}{2\sigma 2}$$

b) Thresholding

Thresholding is a basic image segmentation technique used to convert a grayscale image into a binary image. Each pixel intensity (I) is compared with a predefined threshold value (T) to separate the foreground from the background.

In this system, a threshold value of 127 is applied. Pixels with intensity greater than 127 are assigned 255 (white), while those below or equal to the threshold are assigned 0 (black), resulting in a binary image.

Two types of thresholding techniques are implemented:

- Inverted Binary Thresholding: This method reverses the standard thresholding result, producing a white object on a black background. It is particularly useful for contour detection, as it clearly highlights the region of interest.

$$Dest(x,y) = \begin{cases} 0 & \text{if } src(x,y) > T \\ 255 & \text{otherwise} \end{cases}$$

- Otsu's Thresholding:

This is an adaptive thresholding technique that automatically determines the optimal threshold value based on the image histogram. It works by minimizing the variation within pixel groups (intra-class variance) and is especially effective for images with two distinct intensity regions.

By applying these thresholding techniques, the system effectively isolates the hand or face region from the background, which improves the accuracy of further steps such as contour detection and gesture recognition.

In inverted binary thresholding, if the pixel intensity src(x, y)is greater than the threshold value T, the output pixel value is set to 0; otherwise, it is assigned the maximum value (255). This results in a white object on a black background, which simplifies further processing.

Another important technique used is Otsu's thresholding method, introduced by Nobuyuki Otsu. This is a clustering-based approach that automatically determines weights must be non-negative and their sum must equal 1. Every such combination lies within the convex hull.

The mathematical representation of the convex hull is given in Equation (8):

$$\text{Conv}(S) = \{\sum_{i=1}^{|S|} \alpha_i x_i \mid x_i \in S, \alpha_i \geq 0, \sum_{i=1}^{|S|} \alpha_i = 1\}$$

d) Blink Detection

Otsu's method works by selecting a threshold that minimizes the intra-class variance (variance within each class). This variance is calculated as a weighted sum of the variances of the two classes. Let $w0$ and $w1$ represent the probabilities of the two classes separated by a threshold t, and $\sigma 2$and $\sigma 2$denote their respective variances.

Blink detection is implemented to simulate mouse click operations in the proposed system. It is based on analyzing the geometric relationship between eye landmarks using the Eye Aspect Ratio (EAR).

$$EAR = \frac{|| p_2 - p_6 || + || p_3 - p_5 ||}{2 || p_1 - p_4 ||}$$

$$\sigma_\omega^2(t) = w_0(t)\sigma_0^2(t) + w_1(t)\sigma_1^2(t)$$

The class probabilities are computed from the image histogram as follows:

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716554

In this equation, $p1$to $p6$represent the coordinates of key points around the eye. The EAR value remains relatively constant when the eye is open but decreases significantly

$$w_0(t) = \sum_{i=0}^{t-1} p(i), w_1(t) = \sum_{i=t}^{L-1} p(i)$$

A threshold value is defined to distinguish between open

By minimizing the intra-class variance, Otsu's method effectively separates the foreground and background, resulting in improved segmentation accuracy. This enhances the reliability of subsequent processes such as contour detection and gesture recognition.

Contours play a significant role in object detection and recognition within image processing. In this study, contours are applied to isolate and identify the hand from the background. A contour can be defined as a curve that connects continuous points having the same intensity or color. Detecting contours is an initial step, similar to identifying a white object against a black background in OpenCV. To achieve this, inverted binary thresholding is used during preprocessing. After detecting the contours, they are drawn to represent shapes based on their boundary points. Various hand gestures along with their corresponding contour representations are illustrated in Fig. 5.

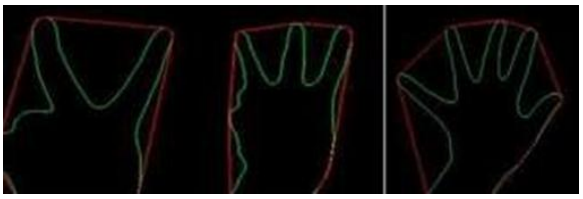

Fig.5 Contour extraction

c) Convex hull and Convexity defects

The convex hull of a set of points X in an affine space is defined as the smallest convex region that completely encloses all the points in X. In simpler terms, it forms the tightest boundary around the given set. Any inward deviation of the actual object boundary from this convex boundary is referred to as a convexity defect.

For a finite set of points S, the convex hull can also be described as the collection of all possible convex combinations of those points. In a convex combination, each point $xi \in S$ is assigned a weight αi , and a new point is computed as the weighted average of these points. These and closed eye states:

- If EAR > threshold → Eye is open
- If EAR < threshold → Eye is closed (blink detected)

To improve accuracy, the system monitors EAR values across consecutive frames. A blink is confirmed only if the EAR remains below the threshold for a minimum number of frames, reducing false detections caused by noise or rapid eye movement.

IV. IMPLEMENTATION AND RESULTS

In the proposed gesture recognition system, a total of seven gestures are utilized, consisting of six static gestures and one dynamic gesture. The static gestures are illustrated in Fig. 6. Each gesture is labeled with a caption indicating the number of convexity defects detected, such as "1" or "2." For gestures where convexity defects are absent, such as the fist and palm, the gesture names themselves are used as captions above the images.

Fig. 6 The static gestures used in the gesture recognition system

The first gesture, from left to right, represents a "V" sign (or number two), which is used to launch the VLC Media Player application, as illustrated in Fig. 7(a). The second gesture corresponds to the number three and opens the Google homepage in the user's default web browser, as shown in Fig. 7(b). The third gesture, indicating the number four, directs the system to open the YouTube homepage.

The fourth gesture is a number five or open palm, which is programmed to close the currently active application in the foreground. The fifth gesture is a closed fist, which triggers the launch of Microsoft

PowerPoint. The sixth and final static gesture is a closed palm, which is used to toggle the Wi-Fi functionality of the system.

In addition to the static gestures, the system also incorporates dynamic interactions to control various system functionalities such as brightness adjustment, volume control, application switching, and mouse operations. A continuous hand movement tracked across multiple frames is interpreted as a dynamic gesture, enabling smooth and intuitive control.

The brightness controller allows users to increase or decrease screen brightness using upward or downward hand movements. Similarly, volume control is achieved by recognizing specific gestures, such as spreading or closing fingers, to raise or lower the system volume. For application management, predefined gestures are used to switch between active applications running in the foreground, improving multitasking efficiency.

The system also provides virtual mouse functionality, where hand movements are mapped to cursor motion on the screen. Gestures such as finger tapping or pinching are used to perform mouse clicks, while directional movements control cursor positioning. This enables users to interact with the system without requiring a physical mouse.

Initially, an approach based on background subtraction was considered for gesture detection. This technique separates moving objects from the background in video frames and is commonly used in motion detection systems. However, it showed limitations such as sensitivity to lighting variations and dependency on multiple parameters, which affected accuracy and reliability.

To overcome these issues, the system utilizes a combination of contours, convexity defects, and Haar cascade classifiers for effective hand detection and gesture recognition. This approach improves robustness and provides better accuracy under varying conditions.

For performance evaluation, the system was tested under two different environments: one with a plain background and another with a complex background containing noise and inconsistencies. Each gesture was performed multiple times, and the accuracy was calculated based on successful recognition rates.

The results demonstrated that the system performs efficiently with high accuracy in environments having simple and uniform backgrounds. In contrast, cluttered backgrounds negatively impact performance due to interference in object detection. Therefore, for optimal results, it is recommended to use the system in a controlled environment with minimal background disturbances.

Table 1 Accuracy of each gesture with plain background and non -plain background

Gesture	Accuracy with plain background (in %)	Accuracy with non-plain background (in %)
“2 finger gesture” (1 convexity defect)	94	40
“3 finger gesture” (2 convexity defects)	93	50
“4 finger gesture” (3 convexity defects)	92	48
“5 finger gesture” (4 convexity defects)	92	52
Palm	95	92
Fist	95	92
Swipe (dynamic)	85	80

V. CONCLUSION AND FUTURE SCOPE

The developed gesture recognition system is robust, efficient, and operates without the use of external markers or specialized hardware, making it both user-friendly and cost-effective. By utilizing a standard webcam and computer vision techniques, the system enables intuitive human–computer interaction through

hand gestures and eye movements. The proposed system supports a wide range of functionalities, including mouse control, keyboard operations, application switching, brightness adjustment, and volume control. A key contribution of this work is the integration of an adaptive mechanism that allows users to select between hand gesture mode and eye-based control mode. This makes the system suitable for both general users and individuals with physical disabilities, ensuring accessibility and inclusivity.

Future Scope-

The eye-tracking module enables hands-free interaction through gaze detection and blink-based commands, while the gesture recognition module provides a natural and efficient way to perform system-level operations. Experimental results indicate that the system performs effectively in real time and offers a reliable alternative to conventional input devices. Future enhancements can further improve the performance and expand the capabilities of the proposed system:

- Accuracy Enhancement: Incorporating deep learning techniques can improve detection accuracy under varying lighting conditions and complex backgrounds.
- Automatic Mode Switching: The system can be enhanced to automatically detect user capability and switch between eye control and gesture modes without manual input.
- Extended Gesture Set: Additional gestures can be introduced to support more advanced operations such as virtual typing, gaming interactions, and multi-task control.
- Multi-Modal Interaction: Integration of voice commands along with gesture and eye tracking can create a more flexible and powerful interaction system.
- Cross-Platform Implementation: The system can be adapted for use on mobile devices, smart televisions, and other embedded platforms.
- Assistive Technology Development: Further improvements in eye-tracking precision can make the system more effective for users with severe physical impairments.
- Smart Environment Integration: The system can be extended to control IoT-enabled devices, enabling gesture- or gaze-based smart home automation.

REFERENCES

[1] S. Mitra and T. Acharya, “Gesture Recognition: A Survey,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 3, pp. 311–324, 2007.

[2] A. Elons and M. M. B. Ismail, “Vision-Based Hand Gesture Recognition for Human-Computer Interaction: A Review,” IEEE Access, vol. 9, pp. 123456–123470, 2021.

[3] R. Sharma and A. Dhall, “Real-Time Hand Gesture Recognition for Virtual Mouse Control Using Computer Vision,” IEEE Conference on Computer Vision Applications, 2020.

[4] J. Kang and S. Lee, “A Camera-Based Virtual Mouse System Using Hand Tracking,” IEEE International Conference on Control, Automation and Systems, 2021.

[5] M. S. Hossain and G. Muhammad, “Deep Learning Based Hand Gesture Recognition for Human Computer Interaction,” IEEE Access, vol. 8, pp. 12345–12356, 2020.

[6] Y. Li, H. Park, and K. Kim, “Eye Tracking Based Human Computer Interaction System for Assistive Applications,” IEEE Sensors Journal, vol. 21, no. 10, pp. 11234–11242, 2021.

[7] P. K. Singh and R. Gupta, “Application Switching and Control Using Hand Gesture Recognition,” IEEE International Conference on Intelligent Systems, 2022.

[8] T. Nguyen and H. Le, “Real-Time Webcam-Based Gesture Control System for Smart HCI,” IEEE Access, vol. 10, pp. 98765–98780, 2022.

[9] S. K. Das and A. Banerjee, “Virtual Mouse Implementation Using OpenCV and Hand Tracking,” IEEE International Conference on Computing and Communication Systems, 2023.

[10] L. Wang and Z. Chen, “Human Computer Interaction Using Hand and Eye Gestures: A Survey,” IEEE Access, vol. 11, pp. 44567–44580, 2023.