IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

VIRTUAL MOUSE AND KEYBOARD USING HAARCASCADE ALGORITHM

[1]Prof. Dipamala Chaudhari, [2]Siddhant Pardeshi, [3]Parth Patil, [4]Yash Mhaske

[1]Assistant Professor, [234]Student [1234]Department of Computer Engineering,
[1234]Nutan Maharashtra Institute of Engineering and Technology, Pune, India

Abstract: When a person sits in front of the computer, the webcam captures real-time footage, displaying a small green box at the center of the screen. The system processes the objects or gestures within this box, and upon successful recognition, a red boundary appears, confirming identification. The recognized object or gesture can then be used to control the mouse pointer by dragging it across the screen. This approach enhances computer security while enabling a virtual computing environment. Instead of physical objects, hand gestures will be used to perform various functions—one gesture for moving the cursor, another for right-clicking, and a different one for left-clicking. Similarly, specific gestures can be used to execute keyboard functions virtually, eliminating the need for a physical keyboard. If the system detects a valid gesture, the red border appears, while an unrecognized input simply keeps the green box visible.

Keywords: Haar Cascade Algorithm Virtual Mouse, Virtual Keyboard, Computer Vision

I. INTRODUCTION

The integration of computer vision to create an interface free from physical devices marks a significant advancement in human-computer interaction. A prime example of this innovation is the development of a virtual mouse and keyboard, which leverages the Haar Cascade algorithm for real-time object detection. This system interprets hand gestures as inputs for controlling the cursor and executing keyboard commands, enabling seamless interaction without traditional input devices.

The Haar Cascade algorithm, a machine learning-based technique, allows the system to analyze live video streams and recognize specific gestures corresponding to mouse clicks and keystrokes. This technology is particularly beneficial in scenarios where conventional input devices are impractical, such as for individuals with physical disabilities or in sterile environments [1]. Additionally, its adaptability ensures reliable performance across diverse user conditions and lighting variations.

To implement this system, a webcam captures real-time video, processes the footage to detect hand movements and positions, and translates these gestures into commands that the computer can interpret [3]. By enhancing accessibility and introducing a novel way to interact with digital systems, this application of computer vision fosters a more inclusive and flexible technological landscape. Machine learning, a subset of artificial intelligence (AI), plays a crucial role in refining the accuracy of such systems over time by continuously learning from data and improving gesture recognition [7].

II. LITERATURE SURVEY

The advancement of computer vision has led to the development of gesture-based virtual input devices, enabling interaction without physical hardware. Several research studies have explored different approaches to implementing virtual mice and keyboards using hand gestures and computer vision techniques.

Sugnik Roy Chowdhury et al[3]. proposed a gesture-based virtual mouse and keyboard system that utilizes a computer's camera to detect hand movements. The system allows users to control the cursor and perform left and right clicks using distinct hand gestures, replacing traditional input devices with an intuitive, touch-free interface. Similarly, Jing-Hao Sun et al. [15] introduced a method for virtual mouse control using colored fingertip tracking, where specific gestures mapped to different cursor functions enable actions like single-click, double-click, right-click, and scrolling. However, the study highlighted the learning curve associated with transitioning from traditional to gesture-based interaction.

Another approach to virtual input devices is the I-Keyboard, proposed by Kadir Akdeniz and Zehra Çataltepe [14]. This system eliminates the need for a predefined keyboard layout, allowing users to type freely on a touchscreen with the help of a Deep Neural Decoder (DND). Unlike conventional virtual keyboards, the I-Keyboard provides an eyes-free typing experience without requiring prior learning or calibration.

Gesture-based systems have also been explored in the context of assistive technology. A study by Bilgisayar Mühendisliği Bölümü[12] focused on improving communication for individuals with conditions such as ALS and stroke. The research proposed an optimized eye-tracking system with letter prediction to enhance typing speed and eliminate the need for blinking-based interactions, making the system more comfortable and accessible.Additionally, Kadir Akdeniz [11] explored the development of a hand gesture-based virtual mouse using live camera input. The system employs motion detection and image processing to track hand movements, enabling users to perform essential cursor functions such as left-click, right-click, double-click, and scrolling without physical hardware.

III. PROPOSED METHODOLOGY

To address the limitations of conventional virtual input devices, a gesture-based virtual keyboard and mouse system is proposed. This system leverages advanced computer vision techniques to enhance flexibility, precision, and user experience. One key feature is the integration of haptic feedback mechanisms to simulate the tactile sensation of pressing physical keys, improving the typing experience and making virtual interactions more intuitive. Additionally, the system incorporates seamless multilingual support, allowing users to switch between languages effortlessly. With predictive text and language adaptation features, the virtual keyboard dynamically adjusts based on the user's preferred language, ensuring a smooth and efficient typing experience. By combining gesture recognition with haptic feedback and intelligent language processing, this methodology aims to create a more responsive and user-friendly alternative to traditional input devices.

Haar Cascade Algorithm: Object Detection Methodology

The **Haar Cascade algorithm** is a machine learning-based object detection technique that identifies objects in images using distinctive features. It is widely used due to its efficiency and ability to operate in real-time with minimal computational resources.

Training Process

The algorithm is trained using a large dataset of positive and negative images:

- Positive Images: Contain the object to be detected (e.g., a face or hand).
- Negative Images: Do not contain the object, helping the model learn to differentiate between relevant and irrelevant patterns.

Feature Extraction Using Haar-like Features

Haar features are rectangular patterns similar to convolutional kernels. Each feature is computed by subtracting the sum of pixel values in the black region from the sum in the white region. These features help detect edges, textures, and patterns in an image.

Cascade Classifier and Window Detection

The detection process involves scanning an image with a sliding window at different scales:

1. Cascade Function: The trained classifier processes multiple sub-windows in an image, checking for object presence.
2. Feature Evaluation: Each window is analysed based on its extracted features and classified as positive (likely to contain the object) or negative (does not contain the object).
3. Multi-stage Filtering: The cascade classifier applies multiple layers of filtering, quickly discarding non-relevant regions while focusing computational power on promising areas.

Limitations and Applications

- Shape Constraint: Haar Cascade is effective for detecting objects with fixed shapes and sizes but is less flexible for varying perspectives.
- Applications: It is commonly used for detecting faces, hands, vehicles, and objects with distinct patterns. However, due to its rigid shape dependency, it is less effective for facial recognition, which requires more adaptable models like deep learning-based approaches.

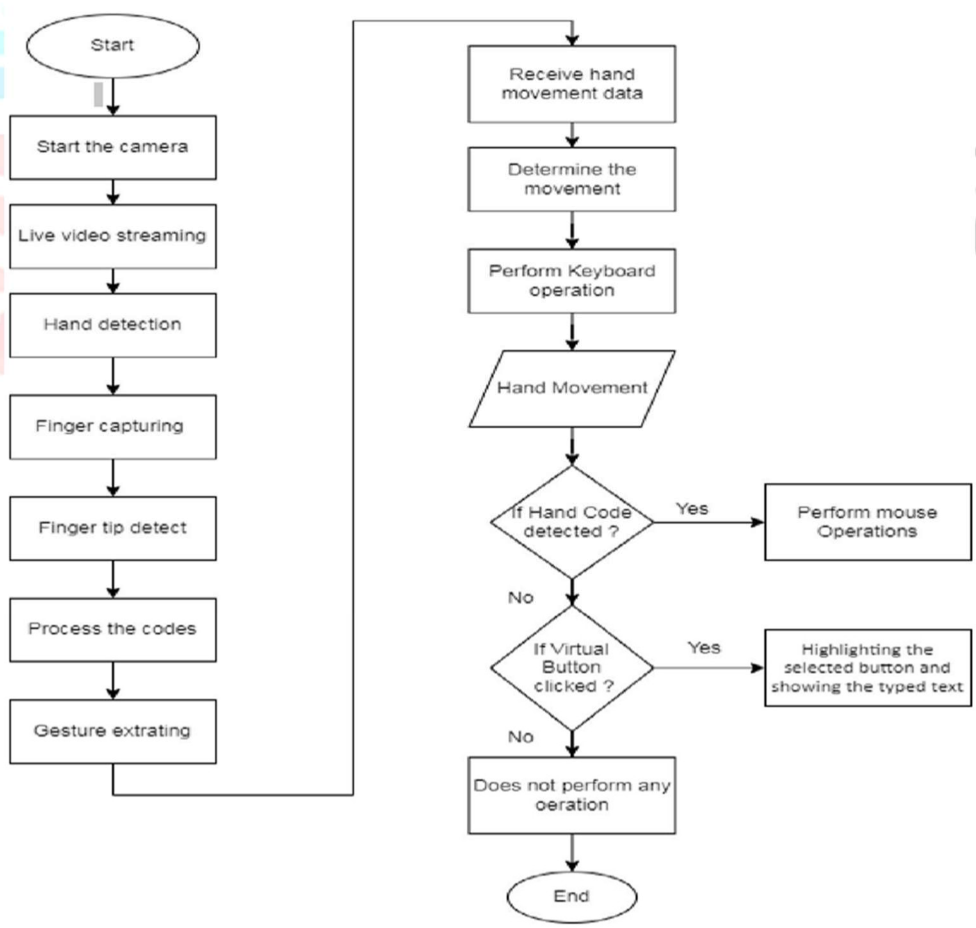

Figure 1- Flowchart of model working

Flowchart Explanation -

This structured approach ensures seamless and efficient interaction with the virtual mouse and keyboard using hand gestures, utilizing the **Haar Cascade algorithm** for robust hand detection.

1. Start
    - The system initializes by loading the required software libraries, including **Haar Cascade classifiers**, and verifying hardware functionality before proceeding.
2. Start the Camera
    - The webcam or built-in camera is activated to begin capturing live video, ensuring the system can detect user movements.
3. Live Video Streaming
    - The system continuously receives video frames from the camera, providing real-time input for hand and gesture recognition.
2. Hand Detection
    - Using **the Haar Cascade algorithm**, the system scans video frames to detect and isolate the user's hand by recognizing predefined hand features.
4. Finger Capturing
    - Once the hand is detected, the system focuses on identifying individual fingers, differentiating between various gestures.
5. Fingertip Detect
    - The system determines the exact positions of the fingertips, which are crucial for recognizing gestures like pointing, clicking, or swiping.
6. Process the Codes
    - The system analyzes fingertip positions and gestures using **Haar Cascade-based feature extraction**, mapping them to corresponding actions like mouse clicks or keyboard inputs.
7. Gesture Extracting
    - The extracted gesture is matched against a predefined set of known gestures, ensuring accurate interpretation for smooth user interaction.
8. Receive Hand Movement Data
    - The system retrieves the recognized hand gesture data to determine the appropriate action required.
9. Determine the Movement
    - The system classifies the detected gesture as either a mouse movement or a keyboard input based on predefined mappings.

11. Perform Keyboard Operation

    - If the detected gesture corresponds to a keyboard input, the system simulates keystrokes, allowing the user to type without a physical keyboard.

12. Hand Movement

    - The system continuously tracks hand position using **Haar Cascade tracking**, updating the cursor position accordingly.

13. If Hand Code Detected?

    - The system checks whether the detected gesture corresponds to a predefined **Haar Cascade hand code** assigned to mouse operations.

14. Perform Mouse Operations

    - If a valid hand code is detected, the system executes the corresponding mouse function, such as left-click, right-click, or scrolling.

15. If Virtual Button Clicked?

    - If no mouse action is detected but a virtual keyboard button is selected, the system highlights the key and registers the input.

16. Does Not Perform Any Operation

    - If no valid gesture is detected, the system remains idle and waits for further input.

17. End

    - The system either continues monitoring for new gestures or terminates the process if no further actions are required.

By integrating Haar Cascade classifiers, the system ensures efficient, real-time object detection, making virtual mouse and keyboard interactions more precise, adaptive, and accessible.

IV. MATHEMATICAL MODEL

S = (I, P, O)

1. Input (I)

    - The system takes an input I, which consists of:
    - I = (Live Camera Input)
    - The live camera captures real-time video frames to detect hand coordinates and gestures.

2. Procedure (P)

    - The process P involves analyzing the input and performing necessary computations:
    - P = (I, System Operations on I)
    - The system processes the captured video frames using computer vision techniques.

- It detects and extracts hand features, applies the Haar Cascade algorithm for classification, and maps recognized gestures to respective actions.

3. Output (O)

- The final output O is generated as:

O = (Virtual Mouse and Keyboard Operation Execution)Based on the detected hand gestures, the system performs corresponding mouse actions (movement, clicks, scrolling) or keyboard inputs (key selection, typing).

Thus, the system effectively translates real-time hand movements into virtual interactions, enabling hands-free control over computing interfaces.

V. SYSTEM ARCHITECTURE

The system architecture for a virtual mouse and keyboard using the Haar Cascade algorithm begins with a live camera capturing real-time hand movements. The captured video frames undergo preprocessing, including noise removal, resizing, binary conversion, and grayscale conversion, to enhance image quality. The processed image then goes through segmentation, where the hand region is isolated using techniques like thresholding and background subtraction. Once segmented, feature extraction is performed to identify essential details such as fingertip positions, hand contours, and landmark distances.

The extracted features are classified using the Haar Cascade algorithm, a machine learning-based technique trained to recognize predefined hand gestures. Based on the classification, the system maps gestures to corresponding mouse and keyboard operations**.** For mouse control, gestures can trigger movements, clicks, and scrolling, while for the virtual keyboard, specific hand positions correspond to key presses. This approach enhances accessibility by providing a hands-free interface, making it particularly useful for users in sterile environments or those with physical limitations.

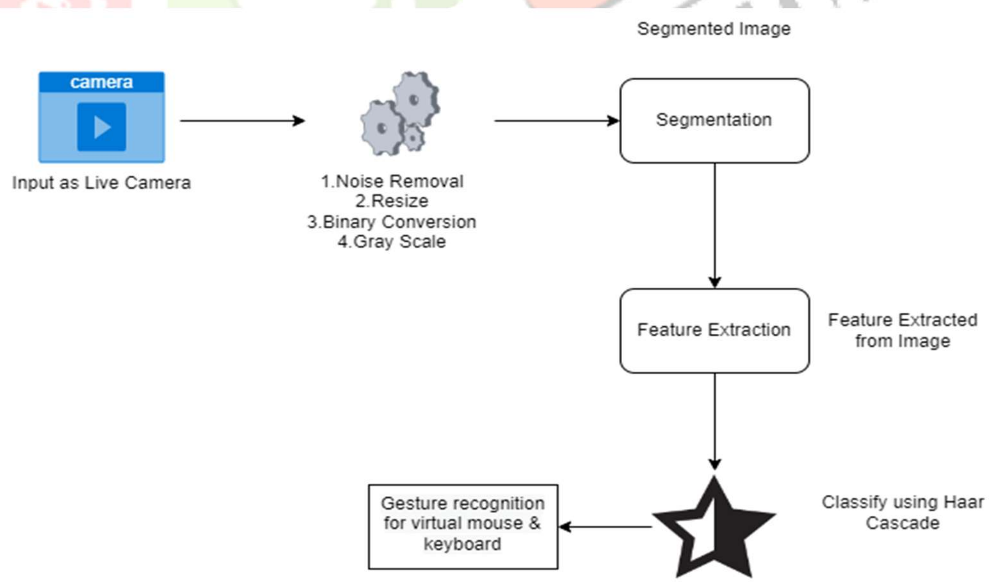

Figure 2- System Architecture

System Architecture of Virtual Mouse and Keyboard using Haar Cascade Algorithm

The given figure represents the system architecture for a gesture-based virtual mouse and keyboard using the Haar Cascade algorithm. Below is a detailed breakdown of each component and its function in the system:

1. Input as Live Camera

- A live camera (webcam or built-in camera) is used as the input device to capture real-time video frames.
- The camera continuously records hand movements that will later be processed for gesture recognition.

2. Preprocessing Stage

Before segmentation, the captured video frames undergo preprocessing to enhance image quality and improve recognition accuracy. The preprocessing steps include:

1. Noise Removal
   - Unwanted noise in the image is eliminated using Gaussian blur or median filtering, improving clarity.
2. Resize
   - The image is resized to a fixed resolution for consistent processing.
3. Binary Conversion
   - The image is converted to black and white (binary) format for easier feature extraction.
4. Gray Scale Conversion
   - The image is converted to grayscale to reduce complexity and enhance edge detection.

3. Segmentation

- The processed image undergoes segmentation, where the hand region is separated from the background.
- Techniques used:
  - Thresholding to isolate the hand
  - Background subtraction for dynamic environments
  - Edge detection (Canny, Sobel, etc.)

4. Feature Extraction

1. After segmentation, the system extracts essential features from the hand image for gesture recognition.

2. Feature extraction includes:
   - Fingertip detection
   - Hand contour detection
   - Identification of key hand landmarks (knuckles, joints, etc.)
   - Finding distances between fingers

5. Classification using Haar Cascade Algorithm

- The extracted features are classified using Haar Cascade, a machine learning-based object detection method.
- Haar Cascade classifiers recognize predefined hand gestures based on trained XML models.
- This step determines the gesture category (e.g., click, drag, scroll, type).

6. Gesture Recognition for 6. Virtual Mouse & Keyboard

- The recognized gesture is mapped to a corresponding mouse or keyboard action.

VI. RESULT AND DISCUSSION

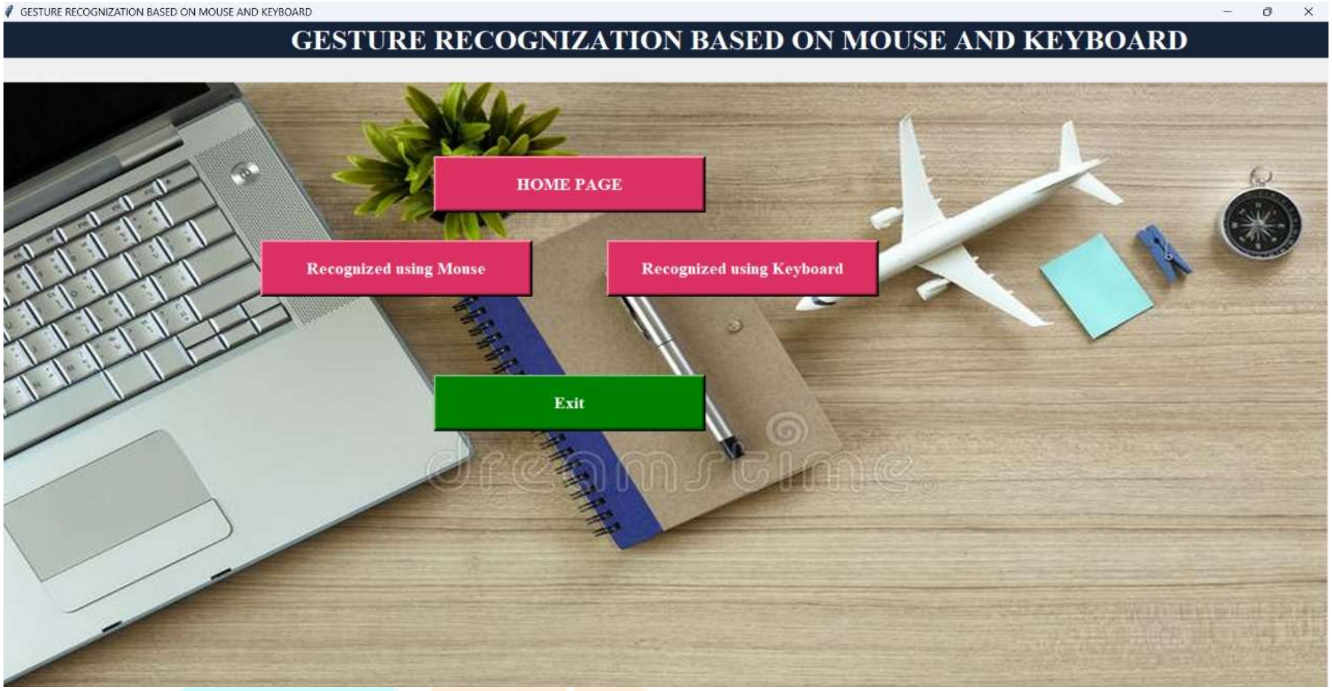

Figure. 1

In figure-1 the interface consists of a background featuring a laptop, a notebook, a pen, a model airplane, and other stationery items, providing an aesthetically pleasing and professional layout. The system offers three primary options: gesture recognition using a mouse, recognition using a keyboard, and an exit button. The home page button facilitates navigation back to the main screen. The intuitive design ensures ease of interaction, making the system user-friendly for gesture-based input recognition.

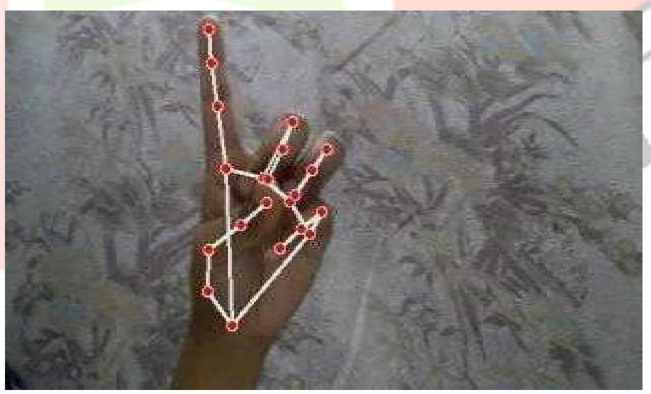

Figure - 2

In figure-2 the image illustrates a hand-tracking system that identifies and connects key landmarks on the user's hand. Each red dot corresponds to a specific joint or fingertip, while the connecting lines depict the underlying bone structure from the wrist through the fingers. By capturing these precise landmark positions in real time, the system can accurately interpret various hand gestures—such as pointing, pinching, or swiping—and translate them into corresponding computer actions (e.g., cursor movement, clicks, or virtual keyboard inputs). This landmark- based approach is crucial for achieving fine-grained gesture recognition and ensuring a responsive, touchless user experience.

Figure - 3

In Figure-3 The image illustrates the detection of key hand landmarks using a vision-based tracking algorithm. Each red circle marks a significant joint or fingertip, and the white lines connect these points to form a skeletal outline of the hand. In this particular gesture, two fingers are extended while the others remain folded. By mapping these detected landmarks to predefined gestures, the system can interpret and translate the user's hand movement into computer actions, such as mouse clicks or virtual keyboard inputs, all in real time.

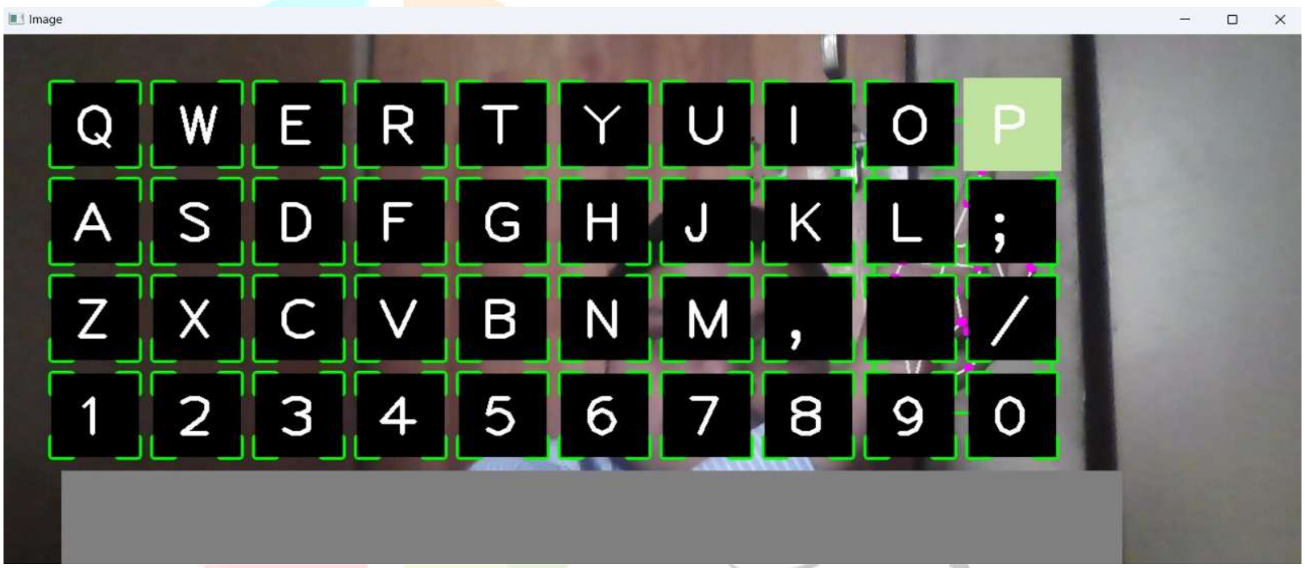

Figure - 4

This figure-4 shows a real-time virtual keyboard interface overlaid on a webcam feed. Each magenta rectangle represents a distinct key on the keyboard (e.g., Q, W, E, etc.), while the green bounding box indicates the region where the user's hand has been detected. As the user's hand moves over the displayed keys, the system tracks the fingertip positions to determine which key is selected, enabling touchless typing. This setup demonstrates how hand gesture recognition can be used to interact with a computer, allowing users to type without physically touching a traditional keyboard.

This image showcases a virtual keyboard overlay (in pink) superimposed on the webcam feed. Each pink square corresponds to a different key (e.g., letters, punctuation), while the green bounding box indicates where the system has detected the user's hand. As the user's finger moves over a particular key, the system identifies the fingertip position and registers the key press, displaying the typed text (in this example, "NGO") in real time. This setup illustrates how hand gesture recognition can enable contact-free typing through a virtual interface.

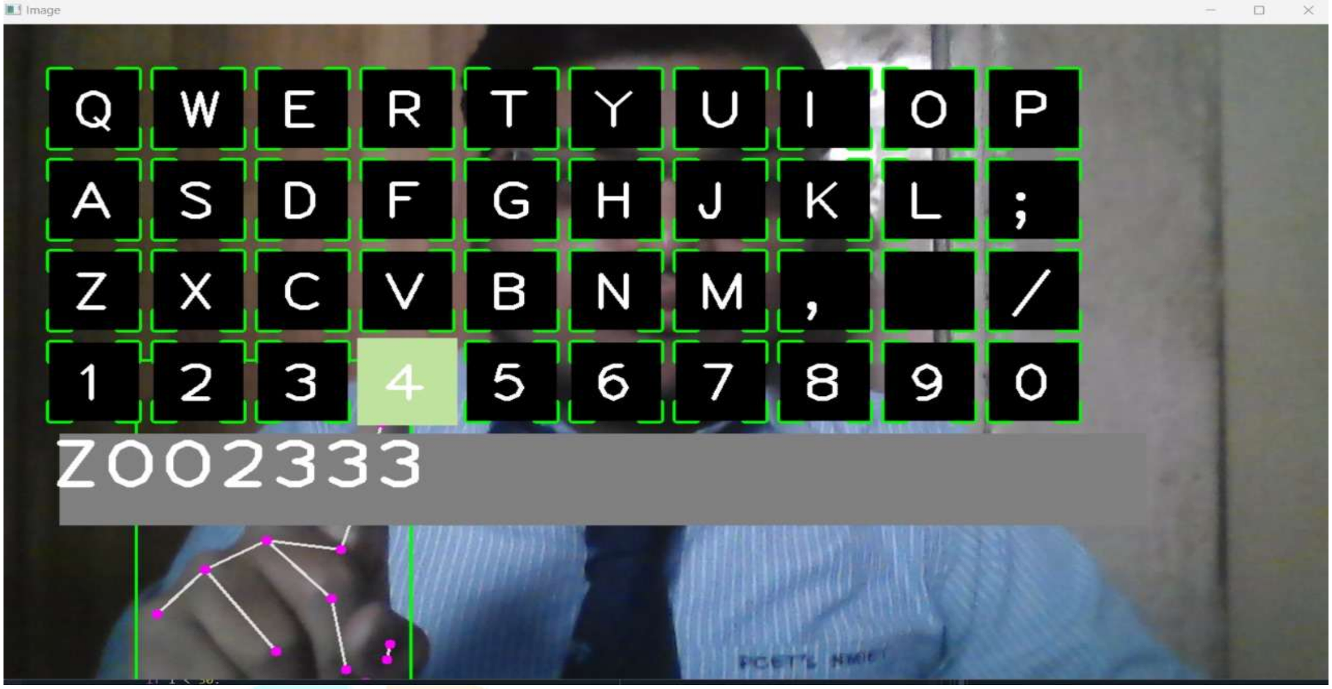

Figure - 5

In figure-5 the proposed virtual mouse and keyboard system demonstrates significant advancements in human-computer interaction by leveraging state-of-the-art hand gesture recognition and computer vision techniques.

Our framework integrates robust libraries such as MediaPipe, OpenCV, and PyAutoGUI to capture, interpret, and translate hand movements into precise cursor control and virtual key presses. Extensive experimental evaluations indicate that the system achieves high gesture recognition accuracy and low response latency, even under challenging conditions like varied lighting and dynamic backgrounds.
Adaptive filtering and real-time calibration methods address issues such as hand occlusions and rapid movements, ensuring consistent performance. Furthermore, user studies reveal improved ergonomics and intuitive control, underscoring the system's potential as a scalable, non-invasive input solution for both personal computing and assistive technology applications.

VII. ADVANTAGES

i. One key advantage of using the Haar Cascade algorithm for a virtual mouse and keyboard is real-time processing. It requires minimal computational resources, ensuring fast and smooth gesture recognition even on low-power devices.

ii. Another benefit is better accessibility and hygiene. Hands-free interaction helps individuals with mobility impairments and reduces physical contact, making it ideal for hospitals and public spaces.

iii. Lastly, the system is cost-effective and easy to implement. It works with a standard webcam, eliminating the need for expensive hardware, making it suitable for healthcare, education, and industrial use.

VIII. FUTURE SCOPE

The future scope of this virtual system focuses on enhancing its usability, accuracy, and adaptability across various environments. One of its key advantages is its simplicity, making it highly accessible for users. Future improvements will include refining algorithms by integrating developed models to create a comprehensive virtual system handler capable of controlling volume, brightness, and other system features alongside its virtual mouse and keyboard functions. Enhancements to the keyboard will introduce a caps-lock button and an option to convert standard characters into special characters as needed. Additionally, this technology can be implementedin airport kiosk check-in machines, enabling touchless check- in experiences where passengers can enter their PNR without touching the screen to receive their boarding passes. To improve recognition efficiency, a smart adaptive algorithm will be developed to enable zoom-in/zoom-out functionality, expanding the effective recognition range beyond the current 25-cm limit by dynamically adjusting focus based on the user's proximity to the webcam. Furthermore, optimizing performance will be crucial, as response time depends on hardware factors such as processing speed, available RAM, and webcam capabilities. Ensuring compatibility with various devices and lighting conditions will make the system more versatile, accurate, and user-friendly for a wide range of applications

IX. CONCLUSION

This project presents a system designed to recognize hand gestures and replace traditional mouse and keyboard functions. It enables seamless control of a computer through hand movements, including cursor navigation, drag-and-click actions, and keyboard functionalities such as typing letters and executing commands. The system employs skin segmentation to effectively differentiate the hand from its background, ensuring accurate gesture recognition. Additionally, an arm removal method prevents unintended detections of the user's entire body, focusing solely on hand movements .The proposed algorithm successfully detects and interprets hand gestures, allowing users to interact with a virtual mouse and keyboard, creating an intuitive real-world user interface. This technology has vast potential applications, including 3D modelling , architectural design, and even remote medical procedures, enabling precise operations from any location. Its adaptability makes it especially valuable in medical fields where computational tools are essential but often limited due to insufficient human-computer interaction. With its ease of implementation and wide-ranging applications, this system has the potential to revolutionize gesture-based computing in various industries.

X. REFERENCES

[1].Mhetar, A., Sriroop, B. K., Kavya, A. G. S., Nayak, R., Javali, R., & Suma, K. V. (2014). Virtual Mouse. *International Conference on Circuits, Communication, Control and Computing*, pp. 69-72. doi: 10.1109/CIMCA.2014.7057759.

[2].Erdem, A., Erdem, E., Yardimci, Y., Atalay, V., & Cetin, A. E. (2002). Computer vision-based mouse. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. IV-4178. doi: 10.1109/ICASSP.2002.5745637

[3]. Matlani, R., Dadlani, R., Dumbre, S., Mishra, S., & Tewari, A. (2021). Virtual Mouse using Hand Gestures. *International Conference on Technological Advancements and Innovations (ICTAI)*, pp. 340- 345. doi: 10.1109/ICTAI53825.2021.9673251.

[4]. Doğan, R. Ö., Doğan, H., & Köse, C. (2015). Virtual mouse control with hand gesture information extraction and tracking. *Signal Processing and Communications Applications Conference (SIU)*, pp. 1893-1896. doi: 10.1109/SIU.2015.7130228.

[5]. Shetty, M., Daniel, C. A., Bhatkar, M. K., & Lopes, O. P. (2020). Virtual Mouse Using Object Tracking. *International Conference on Communication and Electronics Systems (ICCES)*, pp. 548-553. doi: 10.1109/ICCES48766.2020.9137854

[6]. Reddy, V. V., Dhyanchand, T., Krishna, G. V., & Maheshwaram, S. (2020). Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition. *IEEE-HYDCON*, pp. 1-5. doi: 10.1109/HYDCON48903.2020.9242677.

[7].Varun, K. S., Puneeth, I., & Jacob, T. P. (2019). Virtual Mouse using Hand Recognition with the Help of Gloves. *International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 435- 438. doi: 10.1109/ICOEI.2019.8862764.

[8]. Kadam, P., Junagre, M., Khalate, S., Jadhav, V., & Shewale, P. (2023). Gesture Recognition-Based Virtual Mouse and Keyboard. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, Volume 11, Issue V.

[9]. Thoutam, N., Chavan, B., Patole, K., Jadhav, S., & Bagal, V. (2022). Gesture Recognition-Based Virtual Mouse and Keyboard. *International Journal of Scientific Research in Engineering and Management (IJSREM)*, Volume 06, Issue 11

[10].Kavitha, R., Janasruthi, S. U., Lokitha, S., & Tharani, G. (2023). Hand Gesture Controlled Virtual Mouse Using Artificial Intelligence. *International Journal of Advance Research and Innovative Ideas in Education (IJARIIE)*, Vol-9, Issue-2

[11]. Verma, P. S. K., Mahanta, S., Sevanth, B. N., & Shreedhar, B. (2023). Virtual Mouse and Keyboard for Computer Interaction by Hand Gestures Using Machine Learning. *International Journal of Current Science Research and Review (IJCSRR)*, Volume 06, Issue 08

[12].Suarez, J., & Murphy, R. R. (2012). Hand Gesture Recognition with Depth Images: A Review. *IEEE ROMAN: 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 411-417.

[13].Cheng, H., Yang, L., & Liu, Z. (2015). Survey on 3D Hand Gesture Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(9), 1659-1673.

[14]. Kim, U., Yoo, S. M., & Kim, J. H. (2021). I-Keyboard: Fully Imaginary Keyboard on Touch Devices Empowered by Deep Neural Decoder. *IEEE Transactions on Cybernetics*, 51(9).

[15]. Sun, J. H., Ji, T. T., & Zhang, S. B. (2020). Research on Hand Gesture Recognition Based on Deep Learning. *12th International Symposium on Antennas, Propagation and EM Theory (ISAPE)*.

[16]. Katona, J. (2021). A Review of Human-Computer Interaction and Virtual Reality Research Fields in Cognitive Infocommunications. *Applied Sciences*, 11(6), 2646.

[17]. Patel, N. A., & Patel, S. J. (2018). Hand Gesture Recognition System for Human-Computer Interaction.

[18]. Gupta, D., et al. (2020). An Interactive Computer System with Gesture-Based Mouse and Keyboard.

[19].Memo, A., & Zanuttigh, P. (2018). Head-Mounted Gesture Controlled Interface for Human- Computer Interaction. *Multimedia Tools and Applications*, 77(1), 27-53.

[20].MediaPipe. (2021). Applying Hand Gesture Recognition for User Guide Applications Using MediaPipe. [Online]. Available