International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Smart Gesture Based Presenter Using ML Techniques

Mrs. Vagolu Vanaja[1], Harini[2], Kandikonda Neha[3], Madhulika Chepuri[4], Kodhuti Sai Koushik[5]

[1] Assistant Professor, Dept. of CSE-Data Science,ACE Engineering College, India
[2345] B.Tech CSE-Data Science,ACE Engineering College, India
Emails:vanajaace@gmail.com,harinisrinivas112@gmail.com,mailkandikondaneha@gmail.com,madhulikachepuri@gmail.com,kodhutikoushik@gmail.com

A B S T R A C T :

The Smart Gesture-Based Presenter is a system that allows users to control slides using hand gestures without touching any device.It uses a webcam and computer vision to detect gestures in real time.The system is built using OpenCV and CVZone to track hand movements and finger positions.

Each gesture is mapped to a command like next slide, previous slide, draw, zoom, or erase.This touchless method improves presentation control in classrooms and meetings. It makes use of basic machine learning techniques for gesture understanding. The project aims to offer smooth, smart, and interactive slide navigation.

Keywords: Smart Gesture-Based Presenter, Hand Gestures, Computer Vision, Real-Time, OpenCV, CVZone, Presentation Control, Machine Learning

Introduction

This project is about controlling a presentation based on hand gestures. This means you do not have to use a mouse, keyboard, or remote control.
The system uses a webcam to track your hand in real time. It uses OpenCV and CVZone to track your hand moving. There are four things you can do based on the movement - change slides, draw, zoom in/out, or erase.

The Smart Gesture-Based Presenter is built using state-of-the-art technologies—OpenCV, MediaPipe and CVZone amongst others to precisely track hand landmarks and accurately identify gestures used to actuate these gestures. The system allows the user to perform four overall primary actions based on a few specific gestures performed by the user: namely navigate slides, draw/annotate, zoom in, zoom-out, and erase drawings.

Literature Review

Gesture recognition is becoming a popular way of interacting with computers and other devices through 'hands-free' control. For example, in [1], methods such as color tracking and contour detection were used to recognize hand gestures. These methods were known to work well in environments where the lighting was good and the background was clear, and could recognize hand gestures. However, they are limited in performance when the lighting changes or the background is messy. In [2], hand gestures were recognized through the use of Convolutional Neural Networks (CNNs). CNN-based gesture recognition methods did a good job of recognizing gestures; however, CNNs required very large amounts of data and powerful computers, which meant they were difficult to leverage in simple everyday systems.
Rule-based systems, such as the one in [3], recognize hand gestures by relying only on hand landmarks found on the hand, fingers, and palm. Rule-based systems are fast and can be designed and implemented easily, because there is no training required, but they limit the recognition to predefined, fixed number of gestures. In [4], combined methods used tools such CVZone and MediaPipe to track users' hands, to allow users to use hand gestures to advance between presentation slides. However, most of the systems only applied hand gesture control to basic actions associated with advancing a slide.In [5], gesture control methods were found to enhance interactions and make the learning experience more interactive for learners. However, many gesture-controlled systems require special hardware for implementation. The project described in this study demonstrated a method of controlling slide presentations and using annotation, zoom, and erase functions through gestures with only a webcam, with means that it could easily be adopted in classrooms for in-person and online learning situations.

Methodology

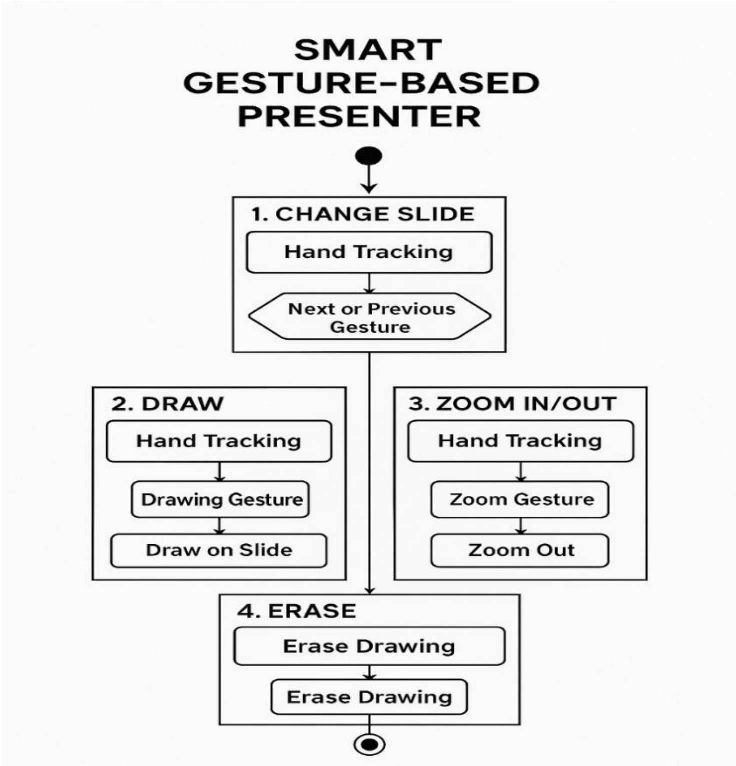


Fig. 1. Smart Gesture Based Presenter Methodology.

Smart Gesture-Based Presenter utilizes real-time hand gestures to enable presenter-controlled, touchless, and intuitive presentation control, using two phases: System Initialization and Gesture-Controlled Interaction, as illustrated in Fig. 1.

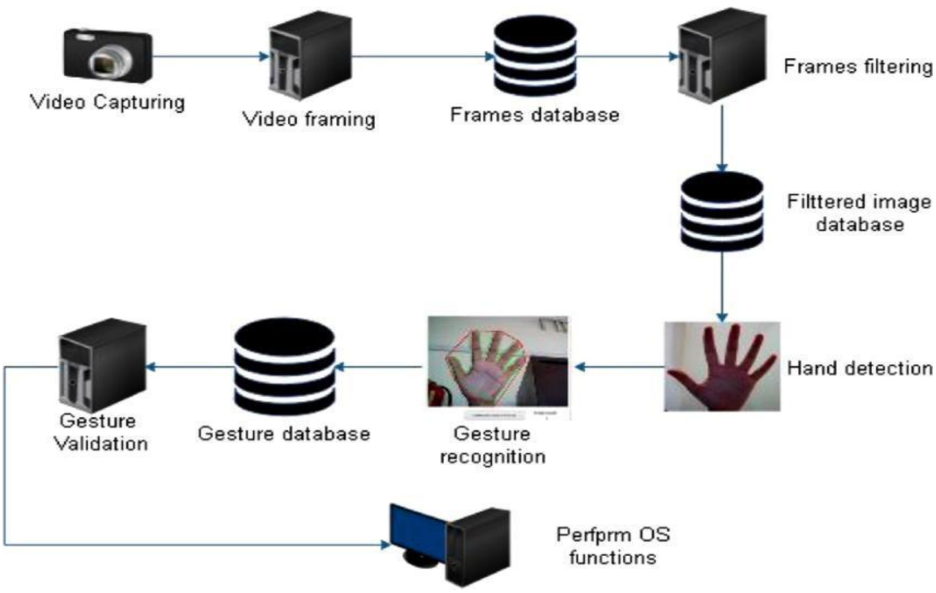
The first part, System Initialization, starts the application and enables the webcam. The application uses CVZone and MediaPipe libraries to identify and track the user's gesture. When the user's hand is identified, a gesture recognition loop begins. The screen displays the current presentation slide and a picture-in-picture, live feed of the user's hand. The display also provides the gesture recognition instructions.

The second part, Gesture-Controlled Interaction, recognizes the gesture and maps the gesture to an action for the presentation. The actions for the presenter are the next slide for the little finger up, previous slide for the thumb up, draw for the index finger up, zoom in for the five finger open, zoom out for all closed fist, eraser for thumb + index. Also, provided to the presenter on the screen is the corresponding action for all recognized gestures.

This type of interaction enables a contactless, engaging, and efficient presenter interaction for educational environments, corporate presentations, and virtual meeting conventions.

System Architecture

Fig. 2. Smart Gesture Based Presenter Architecture



This gesture recognition system is built to detect and understand hand gestures from live video input and use them to perform certain computer functions. The system is divided into several modules, each responsible for a specific task. It is designed to be fast, user-friendly, and suitable for real-time interaction.

1. **Video Capturing**
The system starts by using a camera to capture live video. This is the raw input from the user and is the first step in detecting hand gestures.

2. Video Framing
The live video is broken down into individual frames (images). These frames are stored in a database so they can be processed one by one.

3. Frames Filtering
Out of all the video frames, only the useful ones (clear frames showing a hand) are selected. This step helps reduce noise and improves accuracy.

4. Hand Detection
From the filtered images, the system looks for hands using image processing. If a hand is found, it is sent to the next stage for gesture recognition.

5. Gesture Recognition
This module identifies what gesture is being shown (like open hand, fist, etc.). It compares the detected gesture with known gestures stored in the database.

6. Gesture Validation
Once a gesture is recognized, it is validated by comparing it with saved gesture patterns. This ensures the gesture is correctly identified before taking any action.

7. Perform OS Functions
After validation, the system uses the recognized gesture to perform specific functions on the computer—such as navigating slides, playing media, or other predefined actions.

This system allows users to control their computer using only hand movements, making it ideal for touchless interaction in presentations, education, or smart environments.

Output Screens:

Fig 3. Hand Gesture to move on to the previous slide

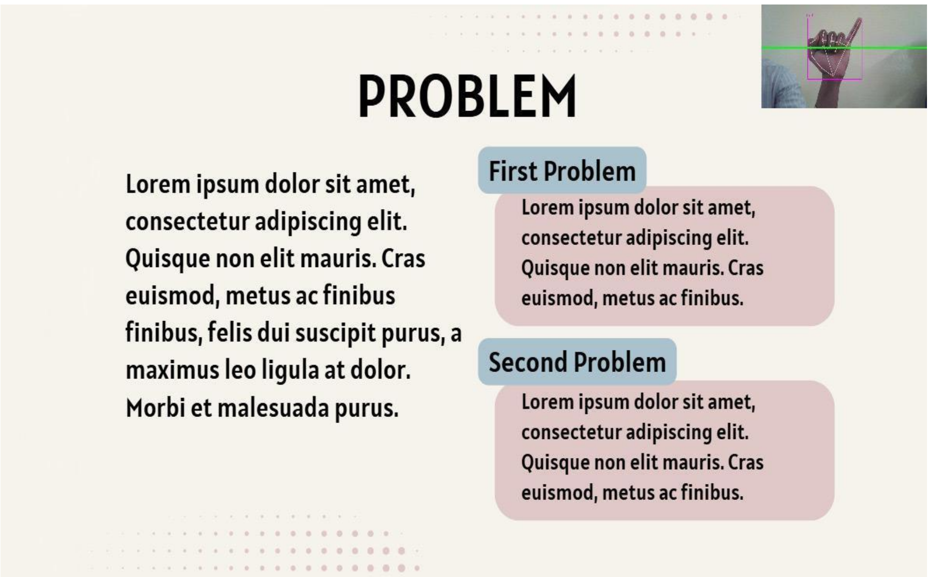


Fig 4. Hand Gesture to move on to the next slide.

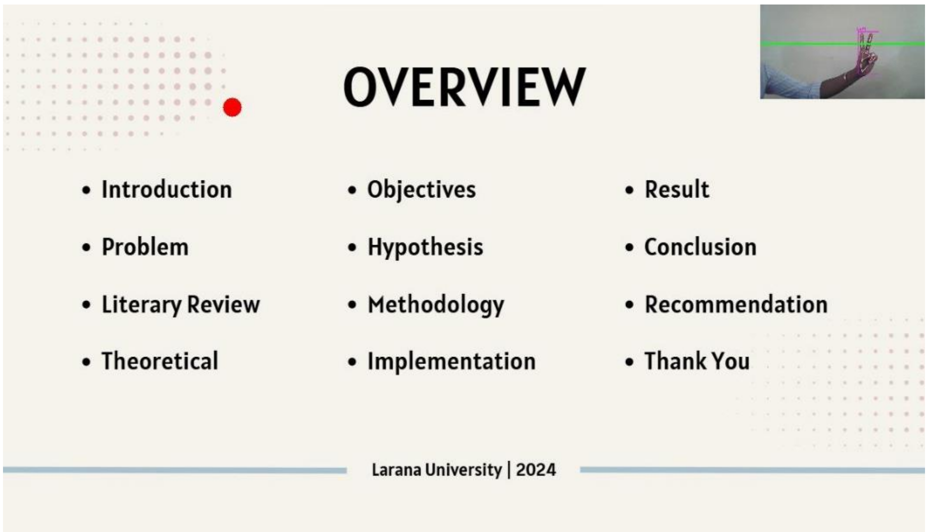


Fig 5. Getting a pointer on slide

Fig 6. Draw using that pointer

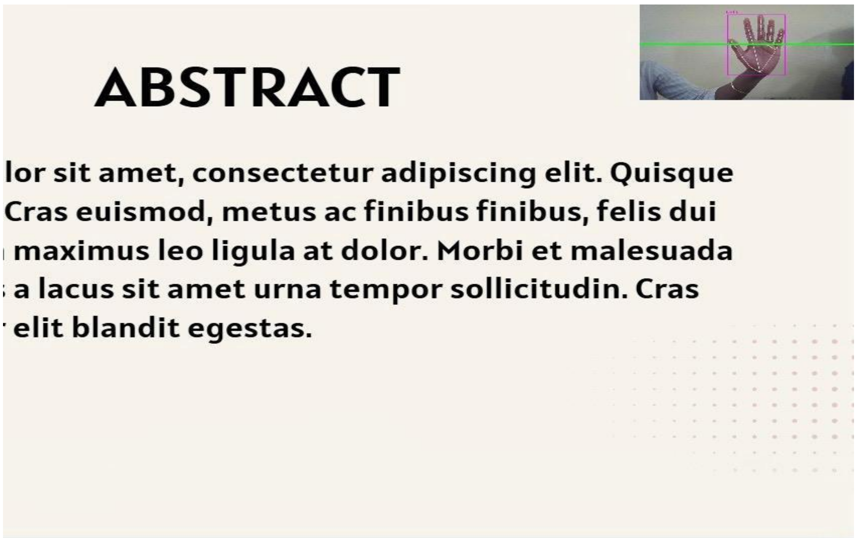


Fig 7. Zooming in the drawing on the slide

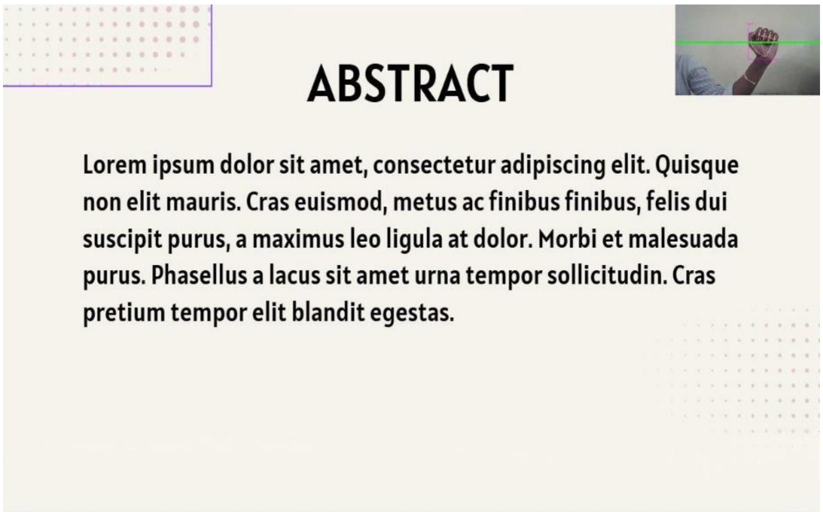


Fig 8. Zooming out the drawing on the slide

Fig 9. Erase the drawing on the slide

Work Flow:

The Smart Gesture-Based Presenter system follows a well-defined workflow that combines webcam-based hand tracking and real-time gesture recognition to deliver a touchless presentation experience. The complete workflow is outlined below:

Step 1: Launch the Application
The user runs the Smart Gesture-Based Presenter Python application, which initializes OpenCV and activates the webcam for live video capture.

Step 2: Webcam Activation
The system opens the webcam and continuously captures live video frames, which are processed in real time.

Step 3: Hand Detection
Using the CVZone and MediaPipe libraries, the system detects the user's hand and identifies its position within the video frame.

Step 4: Landmark Extraction
Key points (landmarks) of the detected hand—such as fingertips, knuckles, and joints—are extracted to understand finger positions.

Step 5: Gesture Recognition
The system analyzes the position of the fingers to identify predefined gestures:

Thumb up → Previous slide

Little finger up → Next slide

Index finger → Drawing mode

All fingers up → Zoom in

Fist → Zoom out

Index finger + Middle finger + Ring finger → Erase mode

Step 6: Map Gesture to Action
Each recognized gesture is mapped to a specific presentation command such as changing slides, annotating, or zooming.

Step 7: Execute Action
The selected action is immediately triggered on the presentation slide, with visual feedback shown in real time.

Step 8: End of Session
The user can exit the application by pressing a key (e.g., 'q'), ending the session and closing the webcam feed.

Conclusion and Future scope:

The *Smart Gesture-Based Presenter* project offers an innovative and touch-free way to control presentations using hand gestures. Instead of relying on traditional input devices like keyboards, mice, or remotes, this system allows the user to move slides, annotate, zoom, or erase content simply by moving their hand in front of the camera. It uses computer vision and hand tracking to recognize specific gestures in real time, making the presentation process more interactive, smooth, and engaging.

This system is especially useful for teachers, professionals, and speakers who want to deliver presentations without needing to stay close to their devices. It reduces physical contact, improves mobility during sessions, and supports a more modern way of interaction. Overall, the project proves to be effective, user-friendly, and a step forward in the field of smart human-computer interaction.

Future Scope:

1. *Improved Gesture Detection Accuracy:*
   Future versions can use advanced deep learning models like *MediaPipe Hands* or *YOLO-based gesture recognition* to better detect hand movements in different lighting and background conditions.
2. *Multi-User Support:*
   The system can be trained to recognize gestures from multiple users, allowing use in classrooms or group presentations where multiple people may want to control the slides.
3. *Voice + Gesture Control:*
   Adding voice commands along with gestures can make the system more versatile and accessible to people with different needs.
4. *Mobile and Web Integration:*

The gesture control system can be linked with mobile or web-based presentation tools (like Google Slides or Microsoft PowerPoint Online) for remote presentation access.

5. *Custom Gesture Configuration:*
   Users can be given the option to set their own custom gestures for specific actions based on personal comfort or presentation style.

6. **Hardware Support for Smart Boards:**
   The system can be integrated with smart boards or digital classrooms, making it a useful tool for smart education and interactive teaching environments.

7. **Real-time Feedback Display:**
   Displaying a small feedback window showing which gesture was detected can help users correct errors and improve gesture performance during live use.

This project has the potential to be developed into a powerful smart presentation tool in schools, colleges, offices, and conferences. With further improvements, it can make digital presentations more intuitive, hands-free, and futuristic.

Acknowledgement:

We express our sincere gratitude to all who supported us throughout the development of this project.We are especially thankful to Prof. Y. V. Gopala Krishna Murthy, General Secretary, and Mrs. M. Padmavathi, Joint Secretary, for providing us the opportunity and environment to carry out this work.Our heartfelt thanks to Dr. P. Chiranjeevi, Head of the Department, for his guidance and encouragement.We are deeply grateful to our internal guide Mr. G. Parwateeswar, and project coordinator Mrs. B. Saritha, for their consistent support, valuable feedback, and motivation throughout the project.

Lastly, we thank all the faculty members, staff for their constant encouragement and support.

REFERENCES:

[1]. S. Rajput, S. Kumar, P. K. Gupta, and S. Verma, "Real-time Hand Gesture Control for interactive presentation using computer vision," International Journal of Innovative Research in Technology (IJIRT), vol. 7, no. 12, pp. 1-7, 2024.

[2]. A. Sharma, S. Goel, and M. Singh, "GESTURE DRIVEN PRESENTATION CONTROL," International Journal of Engineering Science & Advanced Technology (IJESAT), vol. 14, no. 05, pp. 88-92, 2024.

[3]. R. Yadav, A. Sharma, and P. Gupta, "Hand Gesture Controlled Presentation using OpenCV and MediaPipe," International Journal Of Engineering Technology And Management Sciences (IJETMS), vol. 7, no. 4, pp. 143-150, 2023.

[4]. S. V. V. Reddy, S. C. Reddy, S. S. A. Khan, and B. S. Reddy, "Hand Gesture Recognition using CVZONE," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 11, no. 5, pp. c408-c414, 2024.

[5]. S. Zhang, J. Chen, and B. Yu, "MediaPipe Hands: On-device Real-time Hand Tracking," Google Research, 2020. [Online]. Available: https://research.google/pubs/mediapipe-hands-on-device-real-time-hand-tracking/

[6]. J. L. B. Bedregal, G. H. C. F. L. Campelo, and A. C. F. R. Bedregal, "Fuzzy Rule-Based Hand Gesture Recognition," in Artificial Intelligence: Proceedings of the 19th IFIP World Computer Congress, TC 12: AI 2006, Volume 1, 2006, pp. 317-326.