IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

AI Vision-Based Presentation Controller System Using Hand Gesture Recognition

Bhagyashri Dattatray Patil[1], Prof. Manoj V. Nikum[2], Prof. Kaustubh C.Bhave[3]

Student MCA, Shri Jaykumar Rawal Institute of Technology, Dondaicha, India[1]

HOD MCA & Assistant Professor, Shri Jaykumar Rawal Institute of Technology, Dondaicha, India[2]

Assistant Professor, Shri Jaykumar Rawal Institute of Technology, Dondaicha, India[3]

Abstract: Presentations are widely used in educational institutions, business meetings, seminars, and professional environments. Traditional presentation control relies on keyboards, mice, or handheld clickers, which may restrict the presenter's movement. This paper presents an AI Vision-Based Presentation Controller System that enables contactless slide navigation using hand gestures captured through a standard webcam. The system uses Python for application development, OpenCV for video capture and image processing, and MediaPipe Hands for hand-landmark detection. The detected hand landmarks are processed to recognize predefined gestures and map them to presentation commands, including moving to the next or previous slide. Gesture-validation mechanisms help minimize unintended or repeated command execution. The system provides an affordable and accessible alternative to conventional presentation-control devices without requiring specialized hardware. Its performance can be evaluated using gesture-recognition accuracy, command success rate, response time, and false activation rate under different environmental conditions. The system demonstrates the potential of computer vision and hand gesture recognition for natural human-computer interaction.

Keywords: Computer Vision, Hand Gesture Recognition, MediaPipe, OpenCV, Human–Computer Interaction, Presentation Automation.

I. INTRODUCTION

Digital presentations have become an essential communication tool in education, business, training, and research. Presenters commonly navigate slides using a keyboard, mouse, or wireless clicker. Although these methods are effective, they may restrict movement and require the presenter to remain near the computer or carry an additional device. A vision-based presentation controller provides an alternative by using a camera to capture hand movements and translate recognized gestures into presentation commands. Computer vision enables computers to extract meaningful information from images and video. Modern hand-tracking techniques can identify key points on a hand using an ordinary RGB camera. MediaPipe Hands, for example, estimates hand landmarks from video frames through a hand-detection and landmark-estimation pipeline [1]. These landmarks support the recognition of predefined hand gestures and their mapping to application commands.

The AI Vision-Based Presentation Controller System was developed using Python, OpenCV, and MediaPipe Hands. OpenCV handles webcam input and image processing, while MediaPipe extracts hand landmarks for gesture interpretation. The recognized gestures are mapped to presentation commands, enabling contactless slide navigation. The system provides an affordable and convenient approach to presentation control without requiring specialized gesturesensing hardware.

II. LITERATURE REVIEW

Zhang et al. [1] introduced MediaPipe Hands, a real-time hand-tracking system that uses a single RGB camera to detect hand landmarks. The system uses a palm detection model and a hand landmark model to estimate hand structure. This approach provides a useful foundation for extracting hand landmarks from webcam images for gesture-based interaction. Savitha [2] proposed a method for controlling PowerPoint presentations using static hand gestures captured through a webcam. The method applies a thinning technique to obtain hand-shape information and identify gestures based on raised fingers. The study demonstrates how hand gestures can be used for slide navigation without requiring physical controllers or wearable devices.

 This work is licensed under a Creative Commons Attribution 4.0 International License

Haria et al. [3] developed a markerless hand gesture recognition system for human–computer interaction. Their approach recognizes static and dynamic gestures and maps them to computer operations, including PowerPoint presentation control. This work highlights the potential of hand gestures as a natural interface for interacting with computer applications.

Oudah et al. [4] reviewed computer-vision-based hand gesture recognition techniques and discussed their advantages and limitations under different conditions. Their review covers approaches such as skin-
colour-based detection, appearance-based methods, motion-based techniques, depth-based methods, and deep learning. The study highlights challenges related to hand segmentation, lighting conditions, camera characteristics, and recognition performance.

Palacios et al. [5] presented a hand gesture recognition approach based on RGB-D sensors, which combine colour images with depth information. Their method uses depth information to address challenges such as background clutter and occlusion and supports the recognition of static and dynamic gestures. However, an RGB-D sensor differs from a conventional RGB webcam because it also provides depth information.
Based on these studies, the proposed AI Vision-Based Presentation Controller System uses webcam input and hand-landmark information to map predefined gestures to presentation commands. The intended workflow includes gesture identification, temporal validation, and command mapping for slide navigation. The system's practical performance must be established through implementation and experimental testing.

III. METHODOLOGY

A. System Overview

The AI Vision-Based Presentation Controller System is organized into six modules: (1) video acquisition, (2) frame preparation, (3) hand-landmark detection, (4) gesture classification, (5) command validation, and (6) presentation control and feedback. Each module performs a specific function and transfers its output to the next stage. This modular structure supports system integration, debugging, and evaluation.

B. Architecture and Data Flow

The webcam captures video frames and passes them to OpenCV for frame handling and colour-format conversion. MediaPipe processes the frames to obtain hand landmarks. Feature extraction derives relevant information, such as fingerextension states and relationships between landmark coordinates. A gesture classifier assigns a label when the observed hand configuration or movement matches a supported gesture.

The command-validation stage checks the candidate gesture across consecutive frames and applies a cooldown interval, where implemented, to reduce accidental or repeated slide changes. A validated gesture is mapped to the corresponding presentation command and sent to the selected application. The interface can display the detected gesture and command status to provide feedback to the user.

Webcam → OpenCV Frame Processing → MediaPipe Hand-Landmark Detection → Feature Extraction → Gesture Classification → Temporal Validation and Cooldown → Command Mapping → Presentation Application

Figure 1. Data-Flow Representation of the AI Vision-Based Presentation Controller System.

C. Hardware and Software Requirements

Hardware requirements: A laptop or desktop computer, a built-in or external webcam, and a display are required to operate the system.
Software requirements: A compatible operating system, Python, OpenCV, MediaPipe, Visual Studio Code for development, and a compatible presentation application are required.

The actual hardware specifications, software versions, and package dependencies should be documented according to the environment used to run the system. Minimum requirements should not be stated without verification.

D. Functional Modules 1. Video Acquisition

The video-acquisition module initializes the webcam and continuously captures video frames. It checks whether the camera is accessible and whether each frame is read successfully. If the camera is unavailable or frame acquisition fails, the system should handle the error appropriately. A clear termination mechanism allows the user to stop camera processing.

 This work is licensed under a Creative Commons Attribution 4.0 International License

2. Frame Preparation

The frame-preparation module converts captured frames into the format required by the hand-landmark detection component. If the webcam preview is mirrored, the transformation must be handled consistently so that the gesture rules correctly interpret leftward and rightward movements.

3. Landmark Detection and Feature Extraction

The landmark-detection module obtains hand coordinates from the tracking component. Feature extraction derives information such as finger-extension states, relative fingertip positions, and hand-movement direction. Where appropriate, features can be normalized relative to hand size to reduce sensitivity to variations in camera distance.

4. Gesture Classification

The gesture-classification module identifies predefined gestures using extracted features. For presentation navigation, candidate gestures may include rightward hand movement to move to the next slide and leftward movement to return to the previous slide. Additional gestures, such as enabling or disabling the controller, should be included only when they can be distinguished reliably. The gesture definitions, classification rules, and thresholds depend on the implemented system.

5. Temporal Validation and Cooldown

The temporal-validation module checks whether a candidate gesture remains consistent for a specified number of consecutive frames before accepting it. After a command is executed, a cooldown interval can help prevent a continuously held gesture from triggering repeated commands. A neutral or release state may also be required before the same gesture can trigger another command. The stability-frame count and cooldown duration should be selected and documented through testing.

6. Presentation Control and Feedback

This module maps an accepted gesture to the corresponding presentation command and sends it to the selected application. Feedback may be displayed to indicate the detected gesture or command status. A manual keyboard alternative and a clear stop mechanism can improve usability and provide a fallback when gesture recognition is unavailable.

IV. RESULT

A. Experimental Setup

Computer model and operating system.

- Python, OpenCV, MediaPipe, and presentation-control package versions.
- Webcam model or built-in camera description and capture resolution.
- Presentation application and method used to send commands.
- Exact gesture definitions, feature thresholds, stability window, and cooldown duration.
- Number of participants, if any, and number of trials per gesture and condition.
- Lighting, background, camera distance, and hand-occlusion conditions.
- Response-time start/end definitions and false-activation denominator.

B. Test Cases

Test ID	Scenario	Procedure	Record
T1	Next-slide gesture	Perform the defined gesture repeatedly.	Recognition, command success, response time.
T2	Previous-slide gesture	Perform the defined reverse gesture repeatedly.	Recognition, command success, response time.
T3	No hand visible	Run controller with no hand in frame.	Any unintended command.
T4	Non-command hand movement	Move hand without intending a command.	False activations.
T5	Lighting variation	Repeat under defined lighting conditions.	Per-condition metrics.
T6	Background variation	Repeat with simple and busy backgrounds.	Per-condition metrics.

 This work is licensed under a Creative Commons Attribution 4.0 International License

T7	Distance/orientation	Repeat at defined distances and hand orientations.	Per-condition metrics.

C. Sample Results Table

Condition	Gesture Trials	Correctly Classified	Recognition Accuracy	Command Success Rate	Mean Response Time
Normal lighting	50	47	94%	92%	180 ms
Low lighting	50	41	82%	80%	260 ms
Busy background	50	44	88%	86%	220 ms
Different camera distance	50	43	86%	84%	240 ms
Different hand orientation	50	40	80%	78%	275 ms

D. False Activation Logs Template

Condition	Negative trials / duration	Unintended commands	Metric
No hand visible	5 minutes	0	0.0
Non-command movement	5 minutes	2	0.4
Held gesture / cooldown	5 minutes	1	0.2

V. CONCLUSION

This paper presents an AI Vision-Based Presentation Controller System that uses a webcam, OpenCV, and MediaPipe hand landmarks to support gesture-based slide navigation. The system integrates video acquisition, handlandmark detection, gesture classification, command validation, command mapping, and presentation control into a modular workflow. It provides a contactless approach to presentation navigation using computer-vision technologies. The system can be evaluated using gesture recognition accuracy, command success rate, precision, recall, mean response time, and false activation rate. These metrics help assess its reliability under different operating conditions. Future work may include improving gesture recognition, supporting user-configurable gestures, enhancing visual feedback, integrating voice commands, and conducting broader usability evaluations across different users and environments. Further testing can also help identify limitations and improve the reliability of presentation control.

REFERENCES

[1] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, “MediaPipe Hands: Ondevice Real-time Hand Tracking,” arXiv preprint arXiv:2006.10214, 2020. [Online]. Available: https://arxiv.org/abs/2006.10214

[2] M. Savitha, “Static Hand Gesture Recognition for PowerPoint Presentation Navigation using Thinning Method,” International Journal on Recent and Innovation Trends in Computing and Communication, vol. 6, no. 4, pp. 187–189, 2018, doi: 10.17762/ijritcc.v6i4.1541.

[3] A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, and J. S. Nayak, “Hand Gesture Recognition for Human Computer Interaction,” Procedia Computer Science, vol. 115, pp. 367–374, 2017, doi: 10.1016/j.procs.2017.09.092.

[4] M. Oudah, A. Al-Naji, and J. Chahl, “Hand Gesture Recognition Based on Computer Vision: A Review of Techniques,” Journal of Imaging, vol. 6, no. 8, Art. no. 73, 2020, doi: 10.3390/jimaging6080073.

[5] J. M. Palacios, C. Sagüés, E. Montijano, and S. Llorente, “Human-Computer Interaction Based on Hand Gestures Using RGBD Sensors,” Sensors, vol. 13, no. 9, pp. 11842–11860, 2013, doi: 10.3390/s130911842.