iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VII **Month of publication:** July 2025

DOI: https://doi.org/10.22214/ijraset.2025.73370

Hand Gesture Controlled Presentation System

Vaka Tejaswini[1], Prof. Kunjam Nageswara Rao[2]
[1]*Student,* [2]*HOD of IT & CA Dept, Department of Information Technology & Computer Applications, Andhra University College of Engineering, Visakhapatnam, AP.*

Abstract: In the evolving landscape of human-computer interaction, touchless control mechanisms are revolutionizing the way we interact with digital content. This project presents a robust and intelligent Hand-Tracking Based Presentation Control System, engineered using Python, OpenCV, and advanced gesture recognition techniques. The system transforms any standard webcam into a powerful input device, enabling users to seamlessly navigate, annotate, and interact with presentation slides using natural hand gestures.

Key features include gesture-driven slide navigation, real-time annotation in multiple colors, shape drawing (line, rectangle, circle), and dynamic screenshot capture with visual flash feedback. The system incorporates a dual-palette interface—color and shape—accessible via intuitive finger taps, allowing users to switch modes effortlessly. Additionally, a gesture controlled brightness adjustment function enhances visual clarity without manual input. By leveraging the cvzone HandTrackingModule and integrating win32com for PowerPoint interoperability, the application ensures high accuracy and responsiveness.

This solution not only eliminates the need for physical clickers or touch interfaces but also enhances the overall presentation experience—making it more engaging, hygienic, and accessible. Designed with scalability and usability in mind, the system is ideal for educational, corporate, and remote communication scenarios, marking a significant step forward in gesture controlled user interfaces.

Keywords: Hand Gesture Recognition, Computer Vision, Real-Time Control, Screenshot Capture, OpenCV, Human-Computer Interaction

I. INTRODUCTION

In today's fast-paced and technology-driven world, the demand for intuitive and contactless interaction has significantly increased. Especially in professional, educational, and public speaking scenarios, presenters often face interruptions caused by manual interactions with traditional input devices like mice and keyboards. To address this limitation, this project introduces a hand gesture controlled presentation system using computer vision, which enables seamless and natural control over presentation slides and tools through real-time hand gestures. The system uses a standard webcam to detect and interpret hand movements, eliminating the need for physical contact and allowing users to present more confidently and fluidly.

The primary aim of this project is to develop a smart presentation system that responds to various predefined hand gestures to perform operations such as slide navigation, drawing annotations, adjusting screen brightness, capturing screenshots, and selecting shapes and colors. By tracking hand landmarks using computer vision and machine learning models, the system provides real-time responses to gestures, making it a reliable alternative to conventional remote controllers and clickers. This approach not only improves user experience but also supports hygienic and touch-free operation, making it especially relevant in post-pandemic environments.

The entire system is built using Python, with key libraries such as OpenCV for video stream processing and MediaPipe for detecting and tracking hand landmarks. The cvzone library simplifies gesture recognition, while pyautogui and win32com are used for automating interactions with PowerPoint. The project integrates these tools to create a real-time, responsive application capable of executing complex operations through simple hand movements. Additionally, pygetwindow is used to activate and manage the PowerPoint window during presentations.

This system can be applied in a variety of real-world scenarios such as classrooms, lecture halls, corporate meeting rooms, museums, and public information kiosks. It is particularly useful in environments where traditional input devices are inconvenient or unsanitary. Educators can conduct lectures more dynamically, business professionals can deliver polished presentations without distraction, and public spaces can offer interactive displays that users can control without touching shared surfaces. Additionally, this system offers improved accessibility for individuals with mobility impairments.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

II. LITERATURE REVIEW

Human-computer interaction (HCI) has witnessed significant evolution over the past two decades, shifting from traditional input devices to touchless and intuitive interfaces. Among these, gesture-based control systems have gained popularity for their ability to offer seamless interaction using natural body movements. One of the most promising areas within this domain is **hand gesture recognition**, especially in real-time applications such as sign language interpretation, robotic control, and presentation navigation. The growing accessibility of computer vision tools and pre-trained models has contributed to the increased feasibility of such systems.

In their widely cited survey, Rautaray and Agrawal [1] extensively reviewed various hand gesture recognition techniques and classified them based on their interaction mechanisms (vision-based, sensor-based) and implementation complexity. Their work established a foundational understanding of the transition from hardware-intensive systems like data gloves and infrared sensors to camera-based systems. While their study offers a holistic view of gesture technologies, it also hints at the growing need for lightweight and real-time applications.

A study by Tan et al. [2] implemented a real-time hand gesture-based system using contour detection and convexity defects. Their system detected finger counts and simple hand movements, demonstrating the potential for controlling multimedia and GUI interfaces. However, it required consistent lighting and suffered under complex backgrounds, which limited its reliability in practical settings. Another work by Khan et al. [3] explored gesture classification using MediaPipe and OpenCV. Their system accurately detected static and dynamic hand gestures for controlling system volume and mouse navigation. Although this approach enhanced efficiency by eliminating the need for deep learning training, it was restricted to basic gesture recognition and lacked task-specific customization such as drawing or presentation control.

An innovative approach by Jain et al. [4] integrated finger-tip tracking for virtual drawing on screen using OpenCV and color segmentation. The system provided a creative way to write or annotate on screen but lacked gesture variety, user feedback, and real-time application support for tasks like PowerPoint slide control or screenshot capture.

More recently, Singh et al. [5] proposed a hand gesture-controlled system using deep learning and TensorFlow for classifying custom-trained gestures. Though the system allowed for flexible gesture sets and higher accuracy, it involved model training and GPU support, making it less suitable for real-time lightweight applications on standard hardware.

From the reviewed literature, it is evident that most gesture recognition systems focused either on detection accuracy or on basic functionalities like controlling the mouse, detecting fingers, or performing virtual writing. However, few projects have integrated all the necessary features required for interactive presentation control—such as slide navigation, freehand annotation, shape drawing, screenshot capture, and brightness adjustment, all driven by real-time gestures and webcam input.

The present system bridges this gap by combining gesture tracking with utility-rich tools for full presentation interaction. It uses MediaPipe for hand landmark detection, OpenCV for image processing, pyautogui and win32com for system-level control, and introduces advanced gestures like fist-based screenshot, color/shape palette selection, and dynamic ghost shape previews. This makes the system highly usable, lightweight, hardware-friendly, and ideal for real-world educational or professional scenarios where touchless interfaces enhance the experience.

III. METHODOLOGY

The methodology followed for this project is based on a modular computer vision pipeline that integrates gesture recognition with real-time system control and visual feedback. The system captures live video from a webcam, detects hand landmarks using MediaPipe, interprets gestures through logical mapping, and then performs the corresponding presentation task—such as navigating slides, drawing annotations, adjusting brightness, or taking screenshots.

A. *Step-by-Step Process*

1) Video Capture and Preprocessing:
   - OpenCV captures frames in real time from the webcam.
   - The captured frames are resized and flipped for a mirror effect for better hand tracking.

2) Hand Detection and Landmark Extraction:
   - The MediaPipe model, integrated through the cvzone library, detects hands and extracts 21 hand landmarks per frame.
   - Both hands are tracked independently to distinguish between slide control (left hand) and drawing/screen operations (right hand).

3) Gesture Recognition:
   - The positions of specific finger landmarks are used to identify finger status (up/down).
   - Unique combinations of finger statuses are mapped to specific gestures.
   - Distance-based gestures (e.g., pinch for brightness control) are computed using Euclidean distance between landmarks.

4) Gesture-to-Action Mapping:
   - Based on the gesture, the control module triggers:
     - Slide navigation (via pyautogui)
     - Annotation (freehand or shape drawing)
     - Screenshot capture with flash effect
     - Color or shape palette selection
     - Brightness adjustment
   - Debounce timers are used to prevent repeated actions.

5) UI Rendering and Feedback:
   - The rendering module overlays:
     - Annotations (lines, shapes)
     - Color/shape palette
     - Webcam preview
     - Brightness slider
     - Screenshot flash effect
   - Live ghost previews are shown while drawing shapes for better precision.

6) PowerPoint Integration:
   - The integration module uses pyautogui to simulate keypresses (next/previous slide).
   - pygetwindow ensures the PowerPoint window is active during slide control.

B. *Architecture*

The system is designed in a modular format comprising the following components:

1) Video Capture Module: Captures real-time video from the webcam.
2) Hand Detection and Tracking Module: Uses MediaPipe or cvzone to detect hand landmarks.
3) Gesture Recognition Engine: Classifies gestures based on finger states and hand orientation.
4) Action Handler: Maps recognized gestures to specific functions (e.g., slide navigation, drawing, screenshot).
5) User Interface Layer: Displays slides, color palette, shape palette, webcam preview, and flash effect.

Input Design

- Input Device: Webcam
- Input Type: Hand gestures with distinct configurations for each action

Output Design

- Slide updates based on navigation gestures
- On-screen drawing or shapes in real-time
- Visual feedback for shape preview ("ghost shape")
- Screenshot flash animation
- Brightness level indicator

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

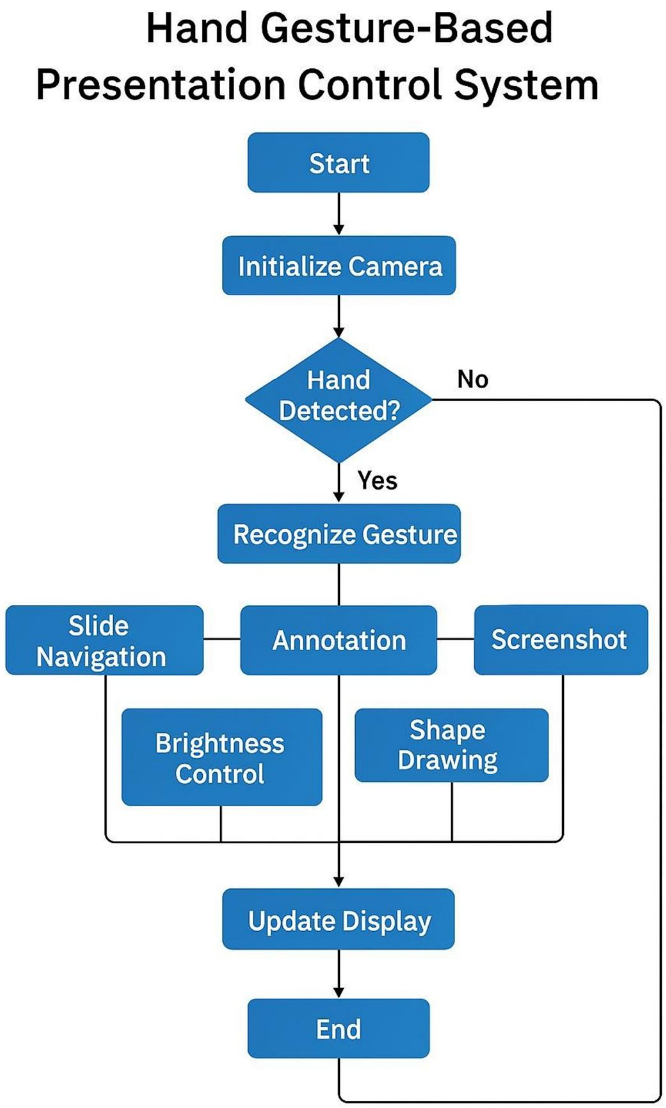


Fig-1 Flowchart of system process

Component	Specification
Processor	Intel Core i5 or higher (or equivalent AMD)
RAM	Minimum 8 GB
Camera	HD Webcam (720p or above)
Display	1366x768 resolution or higher
Storage	Minimum 500 MB of free disk space
GPU *(Optional)*	Dedicated GPU for better frame processing speed

Table-1 Hardware Requirements

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

Software	Version
Operating System	Windows 10/11
Programming Language	Python 3.7 or higher
Libraries/Dependencies	OpenCV, cvzone, mediapipe, pyautogui, numpy, time, math, win32com.client
Presentation Software	Microsoft PowerPoint (optional, for direct slide control)
IDE (for development)	VS Code / PyCharm (recommended)

Table-2 Software Requirements

IV. RESULTS AND DISCUSSIONS

Folder / File	Purpose
gesture_presentation.py	Main script – runs everything
ppt/presentation2/	Contains images of the presentation slides (PNG/JPG format)
ppt/icons/	Icons for palette UI (e.g., red, green color buttons, shape selectors)
ppt/screenshots/	Auto-saves screenshots when the screenshot gesture is performed

Table-3 File details

A. Sample Output

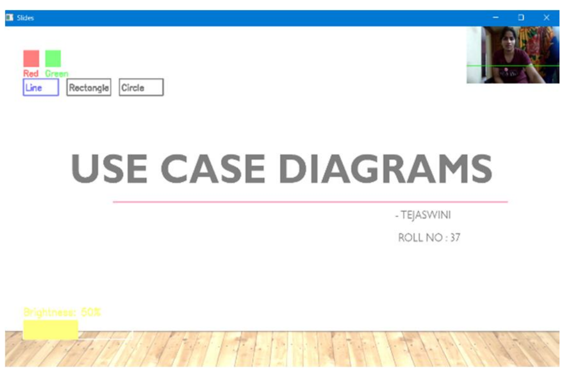


Fig-2 Slide output

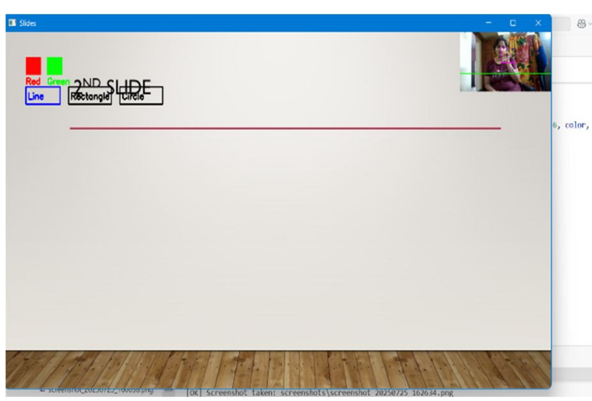


Fig -3 slide navigation using fingers

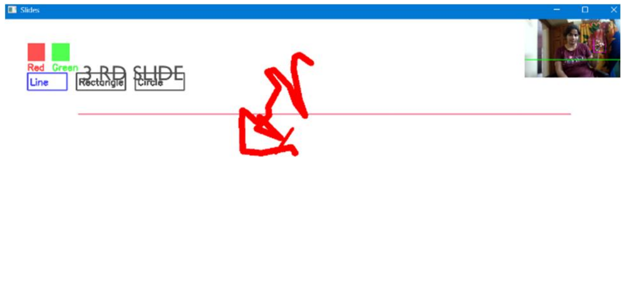


Fig-4 drawing annotations on slide

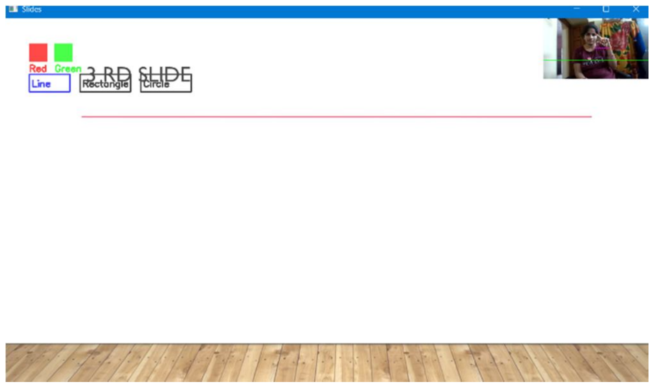


Fig-5 erasing annotations using gestures

```
[OK] Screenshot taken: screenshots\screenshot_20250725_164324.png
[OK] Screenshot taken: screenshots\screenshot_20250725_164441.png
[OK] Screenshot taken: screenshots\screenshot_20250725_164450.png
[OK] Screenshot taken: screenshots\screenshot_20250725_164507.png
[OK] Screenshot taken: screenshots\screenshot_20250725_164519.png
```

Fig -6 creenshot confirmation text

V. CONCLUSION

The Hand Gesture Controlled Presentation System offers a modern, touch-free solution for navigating and interacting with presentation slides using intuitive hand gestures. By leveraging computer vision techniques, OpenCV, and hand tracking models, this system eliminates the need for traditional input devices like keyboards, mice, or clickers. The integration of features such as slide navigation, annotation, shape drawing, screenshot capture, and brightness adjustment enhances user experience and engagement, particularly in professional and academic settings. The system is tested for functional reliability, real-time responsiveness, and user-friendliness across various environments. Overall, this project successfully demonstrates the potential of gesture recognition in improving human-computer interaction during presentations.

The system can be enhanced by integrating AI-based custom gesture recognition, voice-gesture hybrid control, and support for remote presentations via web or mobile platforms. Future improvements may also include multi-user support, more advanced annotation tools, better performance in varying lighting conditions, and accessibility features to assist differently-abled users.

REFERENCES

[1] Rautaray, S. S., & Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: a survey. Artificial Intelligence Review, 43(1), 1–54. DOI: [10.1007/s10462-012-9356-9](https://doi.org/10.1007/s10462-012-9356-9)

[2] Molchanov, P., Gupta, S., Kim, K., & Kautz, J. (2015). Hand gesture recognition with 3D convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 1–7.

[3] Ren, Z., Yuan, J., Meng, J., & Zhang, Z. (2013). Robust part-based hand gesture recognition using Kinect sensor. IEEE Transactions on Multimedia, 15(5), 1110–1120.

[4] Kim, D. H., & Song, W. (2014). A hand gesture recognition method using fingertip detection based on convex hull. International Journal of Advanced Robotic Systems, 11(2), 1–11.

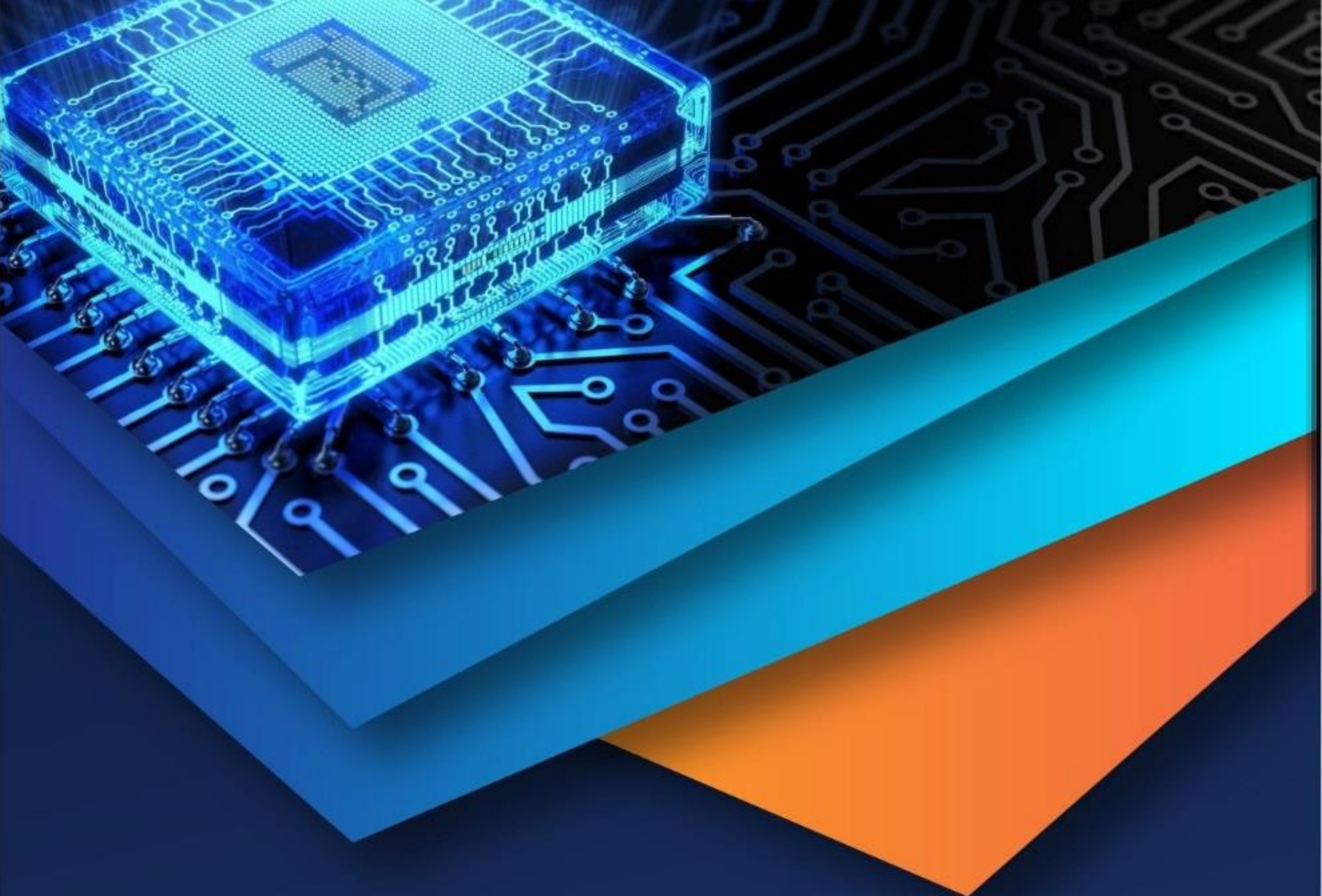
[5] OpenCV Development Team. (2023). Open Source Computer Vision Library (OpenCV). Retrieved from [https://opencv.org](https://opencv.org

[6] Zimmermann, C., & Brox, T.(2017). Learning to estimate 3D hand pose from single RGB images. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 4903–4911.

[7] Mittal, A., & Bhandari, A. (2020). Hand Gesture Recognition using Computer Vision. In International Journal of Scientific & Engineering Research (IJSER), 11(3), 514–518.

[8] Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 677–695.

[9] Google Mediapipe Team. (2022). MediaPipe Hands: On-device Real-time Hand Tracking. Retrieved from https://google.github.io/mediapipe/solutions/hands



10.22214/IJRASET

INDEX COPERNICUS
45.98

ISRA JIF
IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY