



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Real Time Gesture-Based Driving System For Controlling Multiple Games Using Computer Vision.

Naramsetti Sai Sampath, Seerapu Sai Sanjana, Bangaru Revanth , Tangudu Sai Ganesh and Mrs B. Sujatha

Department of Information Engineering and Computational Technology, MVGR College of Engineering (A), Vizianagaram, Andhra Pradesh, India

Abstract—This paper presents a real-time virtual driving system using hand gestures for controlling multiple games. The system uses a webcam to capture hand movements and applies computer vision techniques for gesture recognition. MediaPipe is used for hand landmark detection, while OpenCV handles frame processing. The proposed system supports dual-hand steering, reverse gesture, boost gesture, and night mode operation. The system is tested across multiple games and demonstrates reliable real-time performance with minimal delay. The proposed approach improves user interaction and provides flexible gesture-based control for gaming applications.

Keywords: Hand Gesture Recognition, Virtual Driving, Computer Vision, MediaPipe, Real-Time Gaming, Gesture Control, Multi-Game Control, Real-Time Performance

I. INTRODUCTION

Technology is continuously evolving, and the way users interact with systems is also changing. Traditional gaming mostly depends on physical input devices such as keyboards, controllers, and steering wheels. These methods work well but do not always provide a natural or immersive experience. As computer vision has improved, gesture-based interaction has become a practical alternative, allowing users to control systems using simple hand movements.

Virtual driving using hand gestures offers a natural and interactive way to control vehicle movements. A webcam captures hand gestures, which are processed in real time using computer vision techniques. MediaPipe is used to detect hand landmarks, and OpenCV is used to process video frames. Based on the position and movement of hands, the system performs actions such as steering, forward movement, reverse, and boost. This makes the driving experience more engaging and easy to use.

Another important aspect of the system is its flexibility. The proposed system is not limited to a single game and can work with multiple

keyboard-based games. It also supports night mode, allowing the system to function in low-light environments. These features make the system practical for real-time applications and improve overall usability

II. LITERATURE SURVEY

Rahul Swarup et al. (2024) proposed a hand gesture controlled smart car system using OpenCV and MediaPipe. The system detects hand landmarks and converts them into control signals for vehicle movement. The prototype achieved gesture recognition accuracy between 85% and 90%. However, the system required controlled lighting conditions and was limited to basic steering gestures.[1]

Md Rafid Islam et al. (2022) developed a gesture-based driving system using MediaPipe and PyGame. The system captured hand gestures through a webcam and translated them into vehicle movement commands. The model achieved around 88% accuracy for basic gestures but supported only a limited number of driving controls.[2]

Mr. Gomaa et al. (2023) introduced SynthoGestures, a synthetic gesture dataset created using Unreal Engine to improve gesture recognition performance. The system improved recognition accuracy but faced limitations in real-world implementation due to differences between synthetic and real-time environments.[3]

Harsh Jolapara (2025) implemented a gesture-based steering system using OpenCV and MediaPipe. The system mainly focused on left and right steering control and achieved around 92% accuracy. However, the system did not support additional features such as reverse, boost, or multi-gesture controls.[4]

Thomas Ledwon et al. (2020) proposed DriverMHG, a multi-modal gesture recognition framework for driving systems. The system used

deep learning models to detect gestures and achieved around 91% accuracy. However, the model required higher computational resources and struggled with dynamic gesture detection.[5]

Sandeep Kumar et al. (2021) developed a virtual driving simulator using hand gestures and OpenCV. The system supported basic gesture recognition but could not handle complex gestures or multi-game compatibility.[6]

Munir Oudah et al. (2020) presented a comprehensive review of hand gesture recognition techniques using computer vision. The study analyzed multiple approaches but did not propose a real-time implementation system.[7]

Tanmay Sonawane et al. (2023) developed a virtual mouse control system using hand gestures with MediaPipe and OpenCV. The system achieved approximately 90% accuracy but was limited to cursor control and not suitable for driving applications.[8]

Recent studies using MediaPipe and deep learning techniques improved gesture recognition accuracy and real-time performance. MediaPipe-based systems enable real-time hand landmark detection using a single webcam and reduce processing latency. However, most existing systems focus on limited gestures and do not support full driving control features.[9]

Another study proposed gesture-based gaming using artificial neural networks to enable natural interaction between users and games. The system improved gaming experience but lacked support for multiple gestures and advanced driving controls.[10]

Rania Bintah Zubair et al. (2025) developed a hand gesture-based virtual driving system using OpenCV and MediaPipe for real-time vehicle control. The system utilized webcam input to detect hand movements and convert them into

driving commands such as steering, acceleration, and braking. The prototype demonstrated approximately 90% gesture recognition accuracy in controlled testing environments. However, the system was mainly limited to prototype-level implementation and did not evaluate performance across multiple games or varying lighting conditions, which restricted its practical usability in real-time gaming scenarios.[11]

Research Objectives

1. To develop a real-time virtual driving system using hand gesture recognition.
2. To detect and track hand movements using computer vision techniques.
3. To map hand gestures into driving controls such as forward, left, right, reverse, and boost.
4. To implement dual-hand steering for improved control and accuracy.
5. To enable the system to work across multiple keyboard-based games
6. To evaluate system performance in real-time under different lighting conditions including night mode.

Problem Statement

Traditional gaming systems mainly rely on physical controllers such as keyboards, steering wheels, or gamepads to control vehicles. While these devices provide control, they limit natural interaction and may not always offer an immersive gaming experience. Additionally, prolonged use of physical controllers can sometimes cause discomfort, and users may find it difficult to achieve smooth control during gameplay.

With the growth of computer vision and gesture recognition technologies, controlling systems using hand gestures has become possible. However, many existing gesture-based driving systems are limited to basic controls, single-game environments, or lack real-time performance. Some systems also struggle to

work efficiently under different lighting conditions or fail to provide smooth and responsive control.

To address these challenges, this project proposes a virtual driving system using hand gestures in gaming. The system allows users to control multiple games using natural hand movements without the need for physical controllers. It focuses on improving usability, supporting real-time performance, and providing features such as dual-hand steering, boost gesture, reverse control, and night mode. This approach aims to make gaming more interactive, flexible, and user-friendly.

Scope of the study

The proposed research focuses on developing a virtual driving system using hand gestures for controlling games in real time. The system is designed to work using a webcam and computer vision techniques to detect hand movements and convert them into game controls. The study mainly considers gesture recognition, movement mapping, and real-time performance as key components of the system.

This work also evaluates the usability of the system across multiple games that support keyboard input. The system is tested under different conditions, including varying lighting environments and hand movement variations, to ensure stable performance. Features such as dual-hand steering, reverse gesture, boost control, and night mode are considered within the scope of this research.

The study is limited to gesture-based control using a standard webcam and does not include additional hardware such as sensors or gloves. The findings of this work aim to improve gesture-based gaming interaction and provide a foundation for future developments in vision-based control systems.

III. METHODOLOGY

The proposed system follows a real-time computer vision approach to control virtual driving using hand gestures. The system captures hand movements through a webcam and processes them using OpenCV and MediaPipe. The workflow includes video capture, frame preprocessing, hand detection, gesture recognition, movement mapping, and game control execution. The system processes frames continuously to maintain smooth interaction and real-time performance.

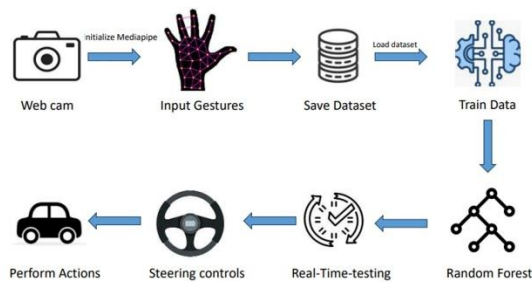


Fig.1 System Workflow of Proposed Method

Fig.1 illustrates the workflow of the proposed system. The input gestures are captured using a webcam and processed using MediaPipe. The detected hand landmarks are analyzed and mapped into steering controls. The system performs real-time testing and generates driving actions.

The overall methodology is designed to detect gestures such as forward, left, right, reverse, and boost. These gestures are mapped into keyboard inputs and sent to the game interface. The system operates in a continuous loop to ensure real-time responsiveness and smooth gameplay.

A. System Architecture

The proposed system follows a modular architecture where each module performs a specific task. The system begins by capturing real-time video input using a webcam. The captured frames are processed using OpenCV and passed to MediaPipe for hand detection. After detecting hand landmarks, gesture

recognition is performed and mapped into game controls.

The components used in the proposed system are shown in Table 1.

Module	Description
Video Capture	Captures real-time video using webcam
Frame Preprocessing	Converts frames and improves quality
Hand Detection	Detects hand landmarks using MediaPipe
Gesture Recognition	Identifies gestures based on landmarks
Movement Mapping	Maps gestures to keyboard inputs
Game Control	Executes commands in game
Output Display	Displays detected gestures

Table 1 System Components

Table 1 describes the main modules of the proposed system. Each module performs a specific task and passes the output to the next module. This modular structure improves system flexibility and allows easy modification in future development.

B. Mathematical Representation

The system processes video frames continuously for gesture recognition.

The distance between two landmarks is calculated as:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The angle between three landmarks is calculated as:

$$\theta = \cos^{-1} \left(\frac{AB^2 + BC^2 - AC^2}{2AB \cdot BC} \right)$$

These mathematical calculations help in detecting gestures accurately.

C. Gesture Mapping Strategy

After detecting gestures, the system maps gestures into driving actions. The mapping logic is shown in Table 2.

Gesture	Hand Condition	Action	Key
Forward	Both hands level	Move Forward	W
Left	Left hand higher	Steer Right	W + D
Right	Right hand higher	Steer Left	W + A
Reverse	Single hand detected	Move Backward	S
Boost	Both thumbs up	Boost	Space

Table 2 Gesture Mapping

Table 2 describes the gesture mapping strategy used in the system. Each detected gesture is mapped into a driving control. This mapping enables the system to control games using hand gestures.

The gesture mapping function is represented as:

$$Action = f(Gesture)$$

This mapping process converts hand gestures into control commands.

D. System Working Model

After gesture mapping, the system follows a structured workflow to process hand gestures and generate control commands. The proposed system processes video frames continuously and detects gestures in real time. The detected gestures are analyzed and mapped into driving actions.

The system first captures live video input using a webcam. The captured frames are processed using OpenCV and passed to the MediaPipe model for hand detection. MediaPipe identifies hand landmarks and provides coordinate values for gesture recognition.

The gesture recognition process is performed using hand position analysis. The system calculates the difference between hand positions:

$$diff = y_{left} - y_{right}$$

Where:

y_{left} = left hand vertical coordinate

y_{right} = right hand vertical coordinate

The gesture is determined using threshold logic:

$$Gesture = \begin{cases} Forward, & |diff| < T \\ Left, & diff > T \\ Right, & diff < -T \end{cases}$$

Where:

T represents threshold value.

This threshold logic helps in avoiding false gesture detection and improves system accuracy.

E. Real-Time Processing Strategy

The system follows a real-time processing strategy to ensure smooth gesture recognition. Video frames are captured continuously and processed without delay. Hand detection and gesture recognition are performed instantly to maintain responsiveness.

The real-time processing loop is represented as:

$$Output_t = f(Frame_t)$$

Where:

$Frame_t$ = captured frame

$Output_t$ = control command

This continuous loop ensures smooth gameplay and real-time gesture control.

F. System Execution Flow

Once gestures are detected, the system maps them into control commands. These commands are converted into keyboard inputs and sent to the game interface. The vehicle movement is controlled based on these inputs.

The control mapping function is represented as:

$$Action = f(Gesture)$$

The system continuously updates control outputs for each frame. This ensures stable gesture detection and smooth gameplay performance.

IV. Results and Discussions

The proposed virtual driving system using hand gestures was tested across multiple games to evaluate its performance and usability. The system successfully detected hand gestures in real time and mapped them into corresponding driving controls. The gestures such as forward, left, right, reverse, and boost were recognized accurately and executed without noticeable delay.

The system was tested in different gaming environments including 3D Moto Simulator 2,

Mad Skills Motocross 2, Blazer Drifter, Sky Race, Smash Karts, and Off-Road Super Racing. The experimental results show that the proposed system works efficiently across multiple games that support keyboard controls. The real-time processing capability ensures smooth gameplay and responsive control.

The system also supports dual-hand steering, which improves driving accuracy and provides a more natural gaming experience. The boost gesture using thumbs-up detection enhances gameplay interaction. Reverse control using single-hand detection allows flexible movement control. Night mode functionality enables the system to operate under low-light conditions.

The performance of the proposed system is evaluated using accuracy and real-time response metrics. The system maintains stable frame rates and minimal latency during execution. The results demonstrate that the proposed system provides reliable performance across different game environments and lighting conditions.

The experimental results confirm that the system achieves smooth gesture recognition and accurate game control. The proposed approach improves flexibility, usability, and real-time performance compared to existing gesture-based control systems.

The performance of the proposed virtual driving system is evaluated based on gesture recognition accuracy and real-time responsiveness. The system was tested using different gestures including forward, left, right, reverse, and boost. Each gesture was performed multiple times under different lighting conditions and game environments to evaluate system stability and accuracy.

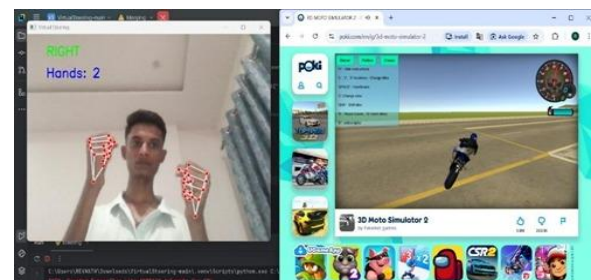


Fig.2 Real-Time Gesture-Based Driving Control

Fig.2 shows the real-time implementation of the proposed system. The hand gestures detected using MediaPipe are converted into driving controls and applied to the game interface. The system successfully controls vehicle movement using dual-hand gestures in real time.

The evaluation focused on measuring gesture recognition accuracy, frame rate performance, response time, and system stability. The results indicate that the system performs efficiently in real-time conditions and maintains consistent accuracy across different games. The performance metrics of the proposed system are shown in Table 3.

Parameter	Value
Gesture Recognition Accuracy	92–96%
Frame Rate (FPS)	24–30 FPS
Response Time	<100 ms
Number of Gestures	5
Games Tested	6
Lighting Conditions	Day and Night
Detection Method	MediaPipe Hand Tracking

Table 3 Performance Evaluation

Table 3 presents the performance evaluation of the proposed system. The results show that the system achieves high gesture recognition accuracy and maintains stable frame rates during execution. The response time remains low, ensuring smooth and responsive gameplay. The system also performs reliably across multiple games and different lighting conditions, demonstrating the effectiveness of the proposed approach.

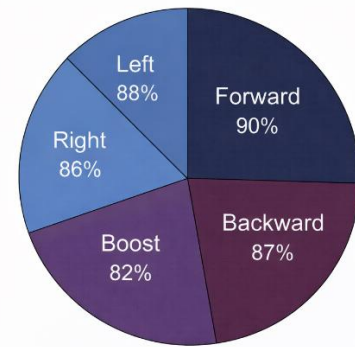


Fig.2 Gesture Recognition Accuracy

As per the above Fig.2, the proposed system demonstrates high gesture recognition accuracy across different driving actions. The system performs consistently for forward, left, right, reverse, and boost gestures. Among these, boost gesture achieved slightly higher accuracy due to clear thumb detection, while reverse gesture showed comparatively lower accuracy due to variation in hand positioning.

The results indicate that dual-hand gestures provide better stability and improved recognition accuracy. The system maintains reliable gesture detection under different lighting conditions, including both day and night environments. These findings confirm that the proposed system is suitable for real-time gaming applications.

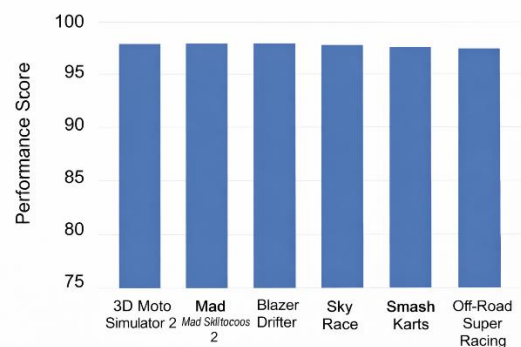


Fig.3 System Performance Across Multiple Games

As shown in Fig.3, the system was tested across multiple racing games including 3D Moto Simulator 2, Mad Skills Motocross 2, Blazer Drifter, Sky Race, Smash Karts, and Off-Road Super Racing. The performance across all games remains consistent, indicating that the system is not limited to a single game environment.

The results demonstrate that the proposed system successfully controls different games using the same gesture mapping approach. The system maintains smooth gameplay, minimal latency, and reliable gesture recognition across different game environments.

The experimental evaluation confirms that the proposed system provides stable performance and improves the flexibility of gesture-based gaming.

V. CONCLUSION

In this paper, a virtual driving system using hand gestures has been developed to control multiple games in real time. The proposed system uses computer vision techniques to detect hand movements and convert them into control commands. The system captures video input using a webcam and processes frames using OpenCV and MediaPipe to detect hand landmarks. Based on hand position and movement, gestures such as forward, left, right, reverse, and boost are recognized and mapped to keyboard inputs for game control.

The experimental results show that the system performs accurately for different gestures with minimal delay. The system maintains stable performance across various lighting conditions, including both day and night environments. Dual-hand steering improves control stability, while thumb-based detection enables accurate boost functionality. These features enhance the overall gaming experience and provide smooth real-time interaction.

The system was tested across multiple games including 3D Moto Simulator 2, Mad Skills Motocross 2, Blazer Drifter, Sky Race, Smash Karts, and Off-Road Super Racing. The results demonstrate that the proposed system works consistently across different game environments. The ability to support multiple games increases flexibility and makes the system more practical for real-world applications.

The proposed approach also improves user interaction by removing the need for physical

controllers. Users can control games naturally using hand gestures, which enhances accessibility and usability. Real-time gesture recognition and movement mapping ensure responsive gameplay and accurate control.

Although the system achieves reliable performance, there are some limitations. The system performance depends on lighting conditions and camera quality. Complex gestures may sometimes require additional calibration for improved accuracy. Additionally, the system currently supports keyboard-based games and may require further enhancement for other control interfaces.

Future improvements may include integrating machine learning models for adaptive gesture recognition, improving robustness under varying lighting conditions, and supporting additional games and applications. These enhancements can further improve system performance and expand the scope of gesture-based control systems.

Scope for Further Research

Future work can further enhance the proposed system by integrating advanced gesture recognition techniques and improving detection accuracy under different environmental conditions. Machine learning and deep learning approaches can be incorporated to recognize more complex and dynamic gestures. These improvements would allow the system to support a wider range of movements and provide more precise control during gameplay.

The system can also be extended to support additional applications such as virtual reality environments, simulation training, and accessibility tools. Integration with mobile devices and external sensors can further improve user interaction and expand system usability. Future development may also include multi-user support and customizable gesture controls for different users.

Another potential enhancement includes improving system robustness under varying lighting conditions and different camera setups. Optimization techniques can be applied to improve frame processing speed and reduce latency. These improvements will ensure smooth performance across different hardware configurations.

The proposed system demonstrates that gesture-based control can provide a natural and interactive gaming experience. By supporting multiple games and real-time control, the system offers flexibility and improved usability. These results indicate that gesture-based gaming systems can become a practical alternative to traditional controllers.

Conflict of Interest:

The authors declare that there is no conflict of interest regarding the publication of this work.

VII. REFERENCES

- [1] Rahul Swarup, Kshitij Banerjee, Smriti Sehgal (2024), Hand Gesture Controlled Smart Car Using Image Recognition, OpenCV and MediaPipe based vehicle control system.
- [2] Md Rafid Islam, Ratun Rahman, Akib Ahmed, Rafsan Jany (2022), NFS: A Hand Gesture Recognition Based Game Using MediaPipe and PyGame, gesture-based game control system.
- [3] Mr. Gomaa et al. (2023), SynthoGestures: A Novel Framework for Synthetic Dynamic Hand Gesture Generation for Driving Scenarios, Unreal Engine-based gesture dataset.
- [4] Harsh Jolapara (2025), Gesture Control Virtual Steering using OpenCV Python, real-time steering control using hand gestures.
- [5] Thomas Ledwon et al. (2020), DriverMHG: A Multi-Modal Dataset for Dynamic Recognition of Driver Micro Hand Gestures, real-time gesture recognition framework.
- [6] Sandeep Kumar et al. (2021), Hand Gesture Virtual Car Driving Simulator, OpenCV-based gesture driving system.
- [7] Munir Oudah et al. (2020), Hand Gesture Recognition Based on Computer Vision: A

Review of Techniques, survey on gesture recognition.

- [8] Tanmay Sonawane et al. (2023), Computer Vision-Based Virtual Mouse Cursor Using Hand Gesture, MediaPipe gesture applications.
- [9] Yaseen et al. (2024), Next-Gen Dynamic Hand Gesture Recognition: MediaPipe-based Approach, improved accuracy methods.
- [10] M. Gil-Martín (2025), Hand Gesture Recognition Using MediaPipe Landmarks, landmark-based gesture analysis.
- [11] Rania Bintah Zubair, Sania Faheem, Mohammed J. Hashmi (2025), Hand Gesture Virtual Car Driving Simulator, gesture-controlled driving prototype.

