

Virtual Driving by Hand Gestures using Open CV and ML

Dasaradha.A¹, K SrinivasRao², Ch Anusri³, P Mohit⁴, H Harika⁵ and D Rohith⁶

¹⁻⁶Aditya Institute of Technology and Management, Tekkali

Email: dasaradha9999@gmail.com

Abstract— Virtual Driving by Hand gestures Using Open CV and aims to presents the development of a virtual steering system for a car racing game utilizing OpenCV, a real-time computer vision library. The primary objective is to create a more intuitive and engaging driving experience by employing image processing techniques to analyse real-world environments and map them to in-game steering inputs. The system utilizes OpenCV to capture video frames from a camera mounted on a physical steering wheel. Image processing algorithms are applied to extract relevant features, such as steering wheel angle and position. These extracted features are then correlated with in-game steering commands, enabling real-time control of the virtual vehicle. To enhance the system's robustness and accuracy, calibration procedures are implemented to account for variations in lighting conditions, camera angles, and individual user characteristics. Additionally, image stabilization techniques are employed to mitigate the effects of camera shake. The main advantage of using OpenCV in a virtual steering system for a car game is the ability to create a more intuitive and realistic driving experience by bridging the gap between the physical and virtual worlds.

Index Terms— Virtual Driving, Virtual Reality, Augmented Reality, Machine learning, Hand Gesture Recognition, Keyboard control, Steering control.

I. INTRODUCTION

Virtual and Augmented Reality (VR/AR) are rapidly evolving technologies that blur the lines between the digital and physical worlds. VR immerses users in entirely digital environments, while AR overlays digital information onto the real world. These technologies have the potential to revolutionize various industries, including gaming, education, healthcare, and architecture, by offering immersive, interactive, and engaging experiences. However, challenges such as technical limitations, high costs, and user experience issues need to be addressed to fully realize the potential of VR and AR.

Hand gesture-based VR gaming offers a unique and immersive experience that transcends traditional controller-based gameplay. By allowing players to interact with virtual environments using natural hand movements, this technology enhances immersion, engagement, and accessibility. It enables innovative gameplay mechanics and facilitates natural social interactions within virtual worlds, pushing the boundaries of interactive entertainment. VR and AR have a wide range of applications, including gaming, education, healthcare, architecture, real estate, retail, automotive, aerospace, and military. These technologies can create immersive, interactive, and engaging experiences, leading to improved learning, enhanced training, and innovative design and development.

OpenCV provides a foundation for general computer vision tasks, enabling video capture, image processing, and result visualization. MediaPipe excels at real-time, high-performance applications, particularly in human-computer interaction. Its hand tracking module accurately detects and tracks key hand landmarks, providing crucial data for gesture recognition. In a hand gesture gaming system, OpenCV handles video input and output, while MediaPipe's specialized hand tracking module delivers the precise data points needed to translate player hand movements into in-game actions.

This synergistic combination of libraries allows for the creation of a robust and responsive system that accurately interprets player gestures for an intuitive gaming experience.

Machine Learning (ML) empowers computers to learn from data, enabling tasks like gesture recognition. Random Forest, an ensemble ML algorithm, excels in classification by building multiple decision trees on subsets of the data, improving accuracy and reducing overfitting. StandardScaler is a crucial data preprocessing step that standardizes features, ensuring that features with larger values do not unduly influence the learning process.

In a hand gesture gaming system, Random Forest can effectively classify hand gestures based on features extracted from MediaPipe's hand tracking data, while StandardScaler enhances the model's performance by ensuring all features are on a similar scale.

II. LITERATURE SURVEY

This survey talks about the contribution of each paper which we have taken as the references. Each paper contains its own priority. Some of the them are as follows.

- Rahman et al. conducted a comprehensive review of hand gesture recognition research, analyzing both non-vision and vision-based approaches. They investigated techniques like HMM, FSM, and color modeling, as well as modern deep learning methods. Vision-based methods, especially those employing deep learning, demonstrated promising results. However, challenges such as real-time processing and robustness to varying conditions remain. Future research should focus on developing more robust algorithms, exploring novel sensors, and addressing the limitations of current approaches to enhance the accuracy and reliability of hand gesture recognition systems.
- Li et al. developed a method for robust hand gesture recognition using deep transfer learning on high-density sEMG signals. They trained a deep convolutional neural network (CNN) model on a source domain and then adapted it to target domains using various transfer learning techniques. Deep transfer learning was employed to bridge the gap between source and target domains in sEMG-based hand gesture recognition. They found that deep transfer learning significantly improved the inter-subject and inter-day performance of the CNN model, outperforming traditional statistical machine learning methods. They concluded that deep transfer learning is a promising approach for robust and adaptable hand gesture recognition, reducing the need for frequent model recalibration.
- Kiran et al. propose a new method for testing autonomous vehicles using virtual reality game simulations. They suggest leveraging the capabilities of consumer VR game engines to create complex and realistic driving environments. They propose a simulation-based approach to test AVs, focusing on the benefits of game engines for creating dynamic and realistic scenarios. They found that VR game simulations offer several advantages over traditional testing methods, including cost-effectiveness, scalability, and the ability to test dangerous scenarios. They concluded that VR game simulations can accelerate the development and validation of autonomous vehicles, making it possible to deploy safer and more reliable AVs.
- Alam et al. developed an immersive VR platform for ASL learning. They used motion capture technology to create a realistic ASL teaching avatar and deep learning to recognize the user's sign language. They combined VR technology, motion capture, and deep learning to create an interactive learning experience. They found that their deep learning model achieved high accuracy in recognizing ASL signs. They concluded that their VR platform can effectively teach ASL and provide real-time feedback to learners.
- Barr et al. identified a gap in current HCI research on video games, arguing that it often overlooks the unique nature of video games as games. They proposed a new perspective on video game HCI, focusing on the concept of "video game values." They adopted a theoretical approach, analyzing the specific characteristics of video games and how they differ from other forms of software. They found that video games possess inherent value systems that influence player behavior and experience. They concluded that a holistic approach to video game HCI, considering both the technical and cultural aspects, is necessary to fully understand and design effective video game interfaces.

II. METHODOLOGY

A. Hand Detection and Landmark Tracking

The system utilizes MediaPipe's hand detection framework to identify and track hand landmarks in real-time through webcam input. The implementation captures 21 key points on each hand, providing precise spatial coordinates for gesture recognition. The system processes video frames at 30 FPS, ensuring smooth and responsive hand tracking while maintaining high detection confidence through the specified threshold of 0.7.

B. Data Preprocessing and Model Training

The system applies a comprehensive data preparation pipeline to take raw landmark coordinates and structure it into feature vectors. StandardScaler normalizes the feature space so that the contribution of all landmarks is uniform. The training process includes real-time data collection and results in immediate feedback such that users can develop personally tailored gesture datasets using an interactive interface along with proper visual guidance.

C. Gesture Feature Extraction and Classification

The system derives features from hand landmark positions to give a feature vector for every gesture. Standardizing these features will then pass to a Random Forest Classifier with 100 estimators, thus accurately classifying already predefined gestures. The training collection process for such data balances user input classes over different gesture classes.

D. Adaptive Steering Control System:

An angle-based control steering mechanism measures the hand orientation from vertical. This has been further divided into three zones: neutral ($\pm 10^\circ$), slight turn ($\pm 30^\circ$), and full turn ($\pm 45^\circ$). During steering transitions, a smoothing factor of 0.5 was applied to give a smooth response and eliminate jitter and sudden movement.

E. Real-Time Control Integration

The system translates detected gestures and steering angles into keyboard commands using the pynput library. It maintains a dynamic key state management system that tracks currently pressed keys and updates them based on new hand positions. The control loop runs continuously, giving immediate feedback through both visual overlays and game control responses.

F. Error Handling and System Robustness

The implementation covers extensive error handling to handle camera failures, data loading errors, as well as runtime exceptions. Hand detection losses and invalid gestures have been accounted for in ways that the system would run with stability in long-playing gaming sessions. Multiple validation checks have been integrated along the processing pipeline to ensure system reliability

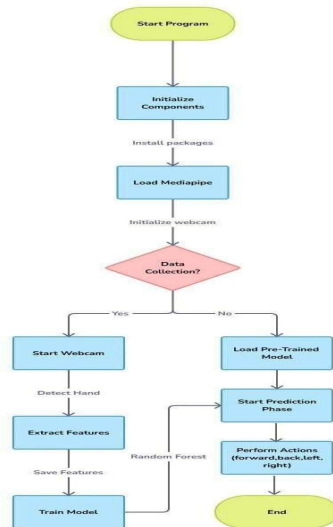


Fig 1

III. RESULTS AND DISCUSSION

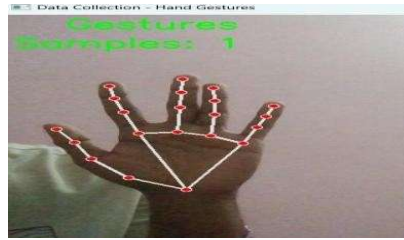


Fig 2

A. Hand Detection and Landmark Tracking

The system utilizes MediaPipe's hand detection framework to identify and track hand landmarks in real-time through webcam input. The implementation captures 21 key points on each hand, providing precise spatial coordinates for gesture recognition. The system processes video frames at 30 FPS, ensuring smooth and responsive hand tracking while maintaining high detection confidence through the specified threshold of 0.7.



Fig 3

By initializing a car racing game, it will ready to race by using the hand gestures, every time a game should be able to handle by the keyboard but now it will ready to race by using our hand gestures.



Fig 4

An angle-based control steering mechanism measures the hand orientation from vertical. This has been further divided into three zones: neutral ($\pm 10^\circ$), slight turn ($\pm 30^\circ$), and full turn ($\pm 45^\circ$). During steering transitions, a smoothing factor of 0.5 was applied to give a smooth response and eliminate jitter and sudden movement. The system translates detected gestures and steering angles into keyboard commands using the pynput library. It maintains a dynamic key state management system that tracks currently pressed keys and updates them based on new hand positions. The control loop runs continuously, giving immediate feedback through both visual overlays and game control responses.

IV. CONCLUSION

The developed hand gesture-based game control system demonstrates that natural and intuitive game interaction via computer vision is feasible. It combines discrete gesture commands with continuous steering control on a dual-zone interface, resulting in a strong and responsive gaming experience. Smooth control with responsiveness

is achieved by implementing smoothing algorithms and adaptive thresholds. Further potential work includes using additional gesture sets and advanced machine learning techniques for further enhancing the capabilities of the system and improving the user experience.

REFERENCES

- [1] Rahman, M. M., Uzzaman, A., Khatun, F., Aktaruzzaman, M., & Siddique, N. (2024). A comparative study of advanced technologies and methods in hand gesture analysis and recognition systems. *Expert Systems with Applications*, 125929.
- [2] Li, J., Jiang, X., Fan, J., Geng, Y., Jia, F., & Dai, C. (2025). Deep end-to-end transfer learning for robust inter-subject and inter- day hand gesture recognition using surface EMG. *Biomedical Signal Processing and Control*, 100, 106892.
- [3] Kiran, A., Salunkhe, S. S., Srinivas, B., Vanitha, M., Ahmed, M. A., & Ramesh, J. V. N. (2024). Accelerating autonomous vehicle safety through Real-Time Immersive virtual reality gaming simulations. *Entertainment Computing*, 50, 100674.
- [4] Alam, M. S., Lamberton, J., Wang, J., Leannah, C., Miller, S., Palagano, J., ... & Quandt, L. C. (2024). ASL champ!: a virtual reality game with deep-learning driven sign recognition. *Computers & Education: X Reality*, 4, 100059
- [5] Barr, P., Noble, J., & Biddle, R. (2007). Video game values: Human-computer interaction and games. *Interacting with computers*, 19(2), 180-195.