WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

HAND GESTURE GAME CONTROLLER USING MACHINE LEARNING

Shreenethi A M[1], Varun S[2], Mrs. R. Jenifer [3]

Sri Shakthi Institute of Engineering And Technology, Coimbatore, TamilNadu, India
E-Mail Id: mohanrajshreenethi@gmail.com, varuns5065@gmail.com

ABSTRACT :

This project presents a real-time hand gesture recognition system designed to control browser-based games such as Subway Surfers using only a webcam. The system eliminates the need for traditional input devices like keyboards or game controllers by leveraging computer vision techniques. MediaPipe is used for accurate hand tracking and landmark extraction, while OpenCV handles video stream processing. Specific gestures—one finger, two fingers, left-hand open, and right-hand open—are mapped to jump, duck, move left, and move right actions, respectively. PyAutoGUI is used to trigger keyboard events corresponding to game controls. The proposed system offers an intuitive, hands-free gaming experience, enabling natural interaction through simple gestures. Experimental results show smooth performance on mid-range hardware, demonstrating the feasibility of gesture-based gameplay.

INTRODUCTION

Human–computer interaction (HCI) has rapidly evolved from conventional devices—keyboards, mice, and controllers—to more natural interfaces. Gesture recognition, powered by modern computer vision techniques, enables users to interact using physical movements, making digital interactions more intuitive.

This project develops a hand gesture–based game controller using a webcam. It integrates MediaPipe for hand tracking, OpenCV for frame processing, and PyAutoGUI for keyboard emulation. Gestures such as finger counting and palm orientation are translated into in-game actions like jumping, ducking, and lateral movement.

Growing interest in immersive and contactless interaction, especially after the pandemic, inspired this system. Unlike specialized devices such as Kinect or Leap Motion, this approach is low-cost and requires no additional equipment. It demonstrates how AI-powered interaction can make gaming accessible, engaging, and device-free.

System Architecture

The system architecture integrates gesture acquisition, processing, classification, and game control to ensure smooth real-time interaction between hand movement and game actions.

Hand Gesture Input Module

This module captures continuous video frames using a webcam. The input is validated, converted to a suitable color format, and passed to the processing layer for landmark detection.

Preprocessing and Frame Handling Module

The captured frames are resized, normalized, and converted to RGB for MediaPipe compatibility. This module ensures smooth real-time processing while handling noise, varying lighting conditions, and hand placement variations.

Hand Landmark Extraction Module

MediaPipe identifies and extracts 21 key hand landmarks for each frame. These landmarks form the basis for gesture classification. The module handles landmark tracking, visibility scoring, and consistency checks to ensure accurate detection.

Gesture Detection Module

Using the extracted landmarks, this module classifies user gestures based on finger count and palm orientation. The following mappings are used:

- One finger → Jump
- Two fingers → Duck
- Left hand open → Move Left
- Right hand open → Move Right

This module translates the movement patterns into logical commands that can be interpreted by the game.

Result Visualization and Annotation Module

PyAutoGUI converts gesture outputs into keyboard events. Simultaneously, annotated frames showing landmarks and detected gestures are displayed to provide real-time visual feedback to the user. This ensures accuracy and improves usability.

Literature review

Gesture recognition has been widely explored due to its potential in improving human–machine interaction.

G.R.S. Murthy & R.S. Jadon emphasize the challenges of real-world gesture recognition, particularly under varying backgrounds and lighting conditions.
Erdem et al. developed a hand-based mouse control system using camera-based image comparison, demonstrating the promise of gesture-driven desktop controls.
Meenakshi Panwar and Pawan Singh Mehra proposed shape-based gesture recognition for visually impaired users, highlighting low computational complexity and ease of implementation.

Pei Xu demonstrated a CNN-based real-time gesture recognition system using monocular cameras, showing high accuracy through effective preprocessing.
Zhang et al. developed a wavelet-based retrieval system, showcasing optimization of computational steps in image-based operations applicable to gesture analysis.
These studies collectively highlight the importance of robust, real-time, and hardware-independent gesture recognition systems—gaps addressed by our proposed approach.

METHODOLOGY

The system follows a structured development methodology to ensure real-time performance and gesture classification accuracy.

1. **Requirement Analysis**

Functional requirements include:

- Real-time hand gesture detection
- Accurate classification of gestures
- Mapping gestures to game actions
- Visual feedback to guide users
- Smooth interaction without hardware dependencies

2. **System Design**

The system is modularized as follows:

- *Image Acquisition Module* – Captures real-time video
- *Preprocessing Module* – Resizes and normalizes input frames
- *Gesture Detection Module* – Uses MediaPipe landmarks for classification
- *Execution Module* – Converts gestures into keyboard events
- *Visualization Module* – Annotates frames with detection results

3. **Technology Stack**
    - Python
    - OpenCV
    - MediaPipe

- PyAutoGUI
- VS Code / PyCharm
- Webcam
- (Optional) NVIDIA GPU for enhanced performance

Integration and Testing

Testing procedures include:

- **Gesture Recognition Tests:** Verifying gesture accuracy under different lighting conditions
- **Frame Processing Tests:** Ensuring smooth frame flow
- **Execution Tests:** Validating real-time game control
- **Visualization Tests:** Checking correct on-screen feedback

Result Analysis

GESTURE → ACTION MAPPING

Gesture	Action
One Finger	Jump
Two Fingers	Duck
Left Hand Open	Move Left
Right Hand Open	Move Right

Figures:

- Fig 7.1 – Jump
- Fig 7.2 – Duck
- Fig 7.3 – Move Left
- Fig 7.4 – Move Right

Conclusion and Future work

The proposed gesture-based game controller proves that natural, AI-driven gameplay is achievable using only a webcam. It eliminates the need for physical controllers and is cost-effective and easy to deploy. The system provides a fully immersive, contactless gaming experience suitable for both entertainment and educational applications.

Future Enhancements:

- Incorporation of deep learning for complex gestures
- Support for multi-hand inputs
- Robust performance in low-light conditions
- Customizable gesture–action mappings
- Extension to VR/AR environments

CONTENT

An intelligent image-based parking management system is proposed that automatically detects This system presents an AI-powered, webcam-based gesture controller that detects hand movements and maps them to game actions. It enhances gaming accessibility, eliminates the need for physical devices, and demonstrates the potential of computer vision in immersive interfaces.

Focus & Scope

Focus	Scope
Hand Tracking	Real-Time Gesture Detection
ML Model	Gesture Classification
Computer Vision	Hands-Free Game Control
Data Processing	Visualization & Control Execution
Scalability	Works Across Browser-Based Games

REFERENCES

1. C. Zhang, J. Yang, C. Southern et al., "Watch Out: extending interactions on a smartwatch with inertial sensing," ACM ISWC, 2016.
2. R. Xie, X. Sun, X. Xia et al., "Similarity Matching-Based Extensible Hand Gesture Recognition," IEEE Sensors Journal, 2015.
3. L. Arouser, P. Bissig, P. Brandes et al., "Recognizing text using motion data from a smartwatch," IEEE PerCom Workshops, 2016.
4. X. Chen, X. Zhang, Z. Zhao et al., "Hand Gesture Recognition Research Based on Surface EMG Sensors," IEEE ISWC, 2007.
5. Suriya R., Vijayachamundeeswari V., "A survey on hand gesture recognition for simple mouse control," ICICES, 2014.
6. Guillaume Plouffe, Ana-Maria Cretu, "Static and dynamic hand gesture recognition using dynamic time warping," IEEE TIM, 2016.
7. Rios-Soria, S. Schaeffer, Garza-Villarreal, "Hand-gesture recognition using computer-vision techniques," 2013.
8. Iason Oikonomidis et al., "Efficient model-based 3D tracking of hand articulations using Kinect."
9. Masanao Yasumuro, Kenya Jin'no, "Japanese fingerspelling identification using MediaPipe.