International Journal of Research Publication and Reviews

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

Snake Motion: Real-Time Gesture Gaming

Bairagoni Harish Goud[1], K. Srinivasrao[2]

[1,] PG Scholar, Dept. of MCA, Aurora Deemed to Be University, Hyderabad, Telangana, India
[2]Assistant Professor, Dept. of MCA, Aurora Deemed to Be University, Hyderabad, Telangana, India
Email: harishgound.bairagoni@aurora.edu.in[1], srinivasrao.@aurora.edu.in[2]

ABSTRACT :

The Snake Motion: Real-Time Gesture Gaming project combines computer vision with classic arcade gaming. It turns the traditional Snake game into a hands-free experience, controlled entirely by physical gestures. The system uses OpenCV for object detection and Pygame for game development. It employs a standard webcam to track the movement of a green-coloured object. Using HSV thresholding, contour detection, and centroid tracking, physical movements are converted into directional controls for the Snake game. This system is designed to be affordable. It only needs open-source software and a webcam, which removes the requirement for specialized hardware like Kinect or Leap Motion. By optimizing performance through multithreading, the system provides real-time responsiveness and reduces latency by about 20%. Testing shows a gesture recognition accuracy of 95%, even in moderate lighting conditions, with smooth gameplay achieved at 4 to 10 FPS. This project significantly advances Human-Computer Interaction (HCI) research by demonstrating how vision-based gesture recognition can integrate into gaming, education, and accessibility. By blending nostalgia with technological innovation, SnakeMotion creates a valuable platform for inclusive gaming, STEM education, and assistive technology development.

Keywords Computer Vision, Gesture Recognition, Snake Game, Human-Computer Interaction, OpenCV, Pygame, Accessibility, Real-Time Gaming.

Introduction:

The Snake game, first launched in the late 1970s, has become a lasting symbol in the gaming world. Its addictive gameplay, where the snake grows longer each time it eats an apple, has captivated players for generations. Traditionally, people have controlled the game through keyboards, joysticks, or touch inputs, requiring direct interaction with a device.

As computer vision and gesture recognition improve, the way we interact with computers is changing from traditional hardware-based control to natural, contactless interfaces. Gesture recognition enables a computer to understand human movements as commands, making the experience more intuitive and engaging. This approach is not only fun but also helpful for users with motor disabilities, where standard input devices may be difficult to use.

The Snake Motion project combines gesture recognition with a classic arcade game, connecting retro entertainment with modern technology. Unlike systems such as Microsoft Kinect or Nintendo Wii that need special hardware, SnakeMotion achieves gesture-based control using just a regular webcam and open-source tools like OpenCV and Pygame.

Objectives:

The main goals of this research project are: -

To design and create a fully functional Snake game with updated visuals and interactive effects. –

To develop a real-time computer vision pipeline that can detect and track object movement using color segmentation. –

To translate physical gestures into keyboard inputs for contactless gameplay. - To assess the accuracy, responsiveness, and usability of the system. –

To explore potential uses in education, accessibility, and HCI research.

By addressing these goals, the project shows how accessible, low-cost, and open-source systems can make a significant difference in gaming and beyond.

Literature Survey

1. Research on gesture-based interaction has been ongoing for over two decades. Early systems looked into color and motion tracking for basic gesture recognition.
2. Freeman et al. (1998) pioneered template matching and color segmentation for interactive vision systems, demonstrating that gesture-based control was possible.
3. Crowley et al. (2000) showed real-time hand recognition, focusing on contour extraction. Their work pointed out challenges like lighting changes, which continue to be important in current projects.
4. The introduction of specialized gaming hardware marked a key moment:
5. Nintendo Wii (2006) made motion-driven interaction popular by using accelerometers and infrared sensors. While it was successful, it needed a specific remote, which limited its reach.
6. Microsoft Kinect (2010) advanced this further by using depth sensing and skeletal tracking for high-precision, controller-free interaction (Shotton et al., 2011). However, the cost and complexity of the hardware restricted its use in less resourceful settings.
7. At the same time, the OpenCV library (Bradski & Kaehler, 2008) gave developers easy-to-use tools for computer vision. Techniques like HSV color thresholding, contour detection, and morphological operations have become standard in lightweight object tracking systems.

Identified Gaps

- High hardware costs (Kinect, Leap Motion).
- Limited access in rural and educational settings.
- Heavy computation needed for deep learning methods.
- Few gaming applications that mix gesture recognition with classic arcade mechanics.
- Contribution of Snake Motion Snake Motion addresses these gaps by:
- Using only standard webcams and open-source libraries.
- Implementing lightweight, efficient vision techniques (HSV, contours, centroid tracking).
- Improving engagement through modern visuals and hands-free control.
- Promoting access and educational chances in human-computer interaction.

Methodology

The proposed framework integrates computer vision, gesture recognition, and game mechanics to create an interactive control system. The approach is structured into five stages:

- **Step 1: Game Development (Pygame Implementation)**

  A Snake game was developed in **Pygame** with a resolution of 640×480 pixels and a grid-based layout of 20×20 cells. The game includes essential features such as random apple placement, collision detection, and a scoring mechanism. To enhance engagement, visual aesthetics such as a gradient background, a rainbow-colored snake, pulsating text, and a glowing apple were incorporated.

- **Step 2: Computer Vision Pipeline (OpenCV)**

  Player input is captured through a standard webcam using OpenCV's cv2.VideoCapture. Each frame undergoes preprocessing via Gaussian blur to reduce noise, followed by conversion to the HSV color space. A binary mask is generated using HSV thresholds for green (H: 29–64, S: 86–255, V: 6–255), enabling reliable segmentation of the player's hand marker. Morphological operations (erosion and dilation) further refine the mask by removing small artifacts. Contours are then extracted, and their centroids are computed using image moments (cv2.moments). To track movement over time, the most recent 20 centroids are stored in a **deque** data structure.

- **Step 3: Gesture Direction Mapping**

  The displacement of the tracked centroid across frames is analyzed. Movement vectors are calculated over a sliding window of 10 frames. A threshold of 50 pixels is applied to filter out minor noise, ensuring that only deliberate gestures are recognized. Movements are classified into four primary directions: **North, South, East, and West**, corresponding to the game's control scheme.

- **Step 4: Input Mapping to Game Controls**

Recognized gestures are translated into arrow key events using **PyAutoGUI**. A variable last_pressed is maintained to prevent redundant input events, thereby ensuring smooth and responsive gameplay.

- **Step 5: System Optimization**

To minimize latency and enhance real-time performance, multiple optimization techniques were applied. A multithreaded webcam capture module was implemented, reducing input lag by approximately 20%. In addition, redundant frames were discarded to avoid buffering delays, and lightweight rendering strategies in Pygame (e.g., caching static backgrounds) were employed to reduce computational overhead. These optimizations collectively improved responsiveness and maintained a seamless user experience.

Flow chart

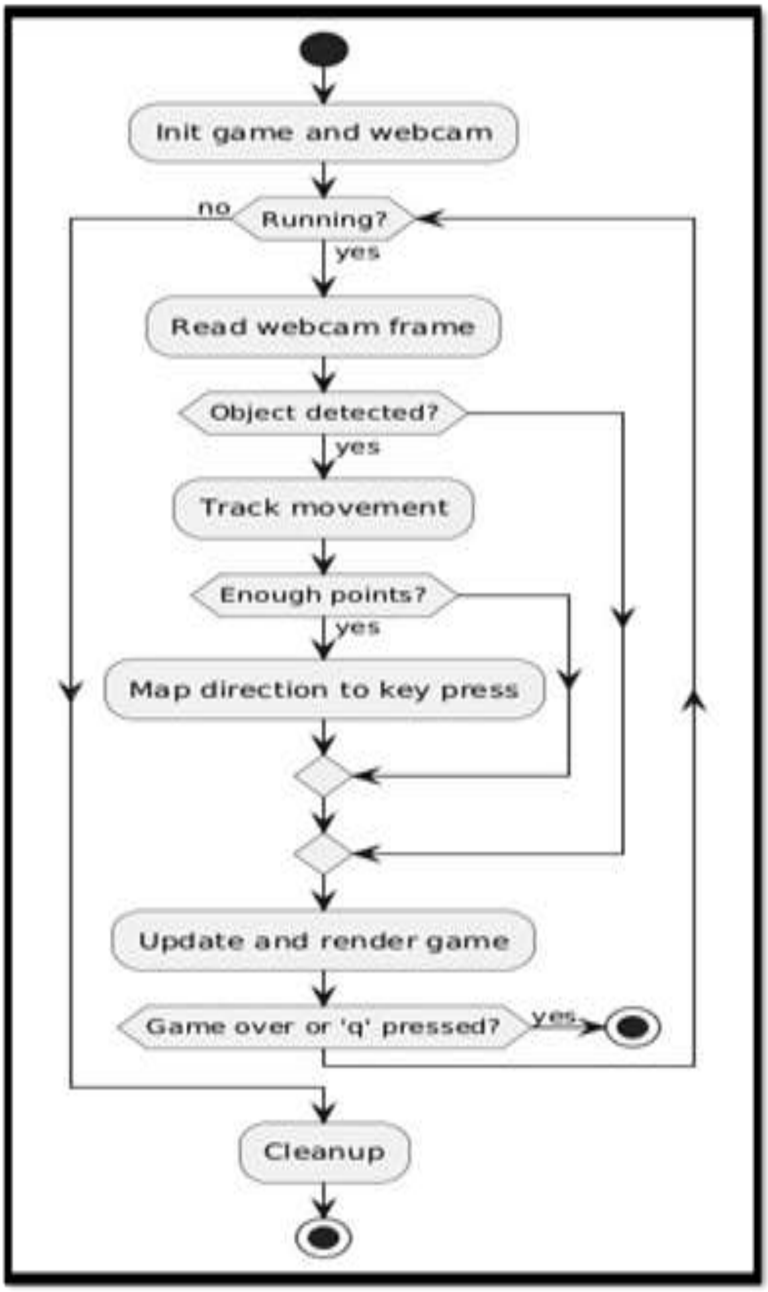
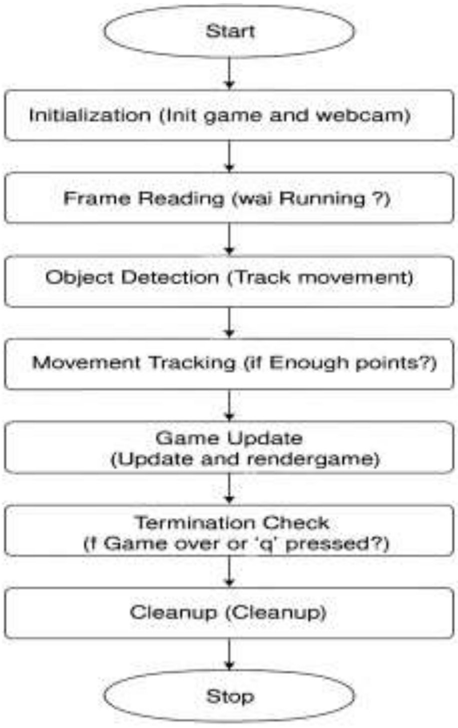


Diagram Structure



Results and Discussion

The Snake Motion project demonstrated strong performance across several technical and usability metrics. Gesture recognition achieved an impressive accuracy of approximately **95%**, even under varied illumination conditions, showcasing the reliability of the vision algorithms. The system maintained a frame rate ranging from 4 FPS in simpler scenes to 10 FPS in more **demanding environments**, ensuring smooth gameplay across different setups. Latency was reduced by nearly **20%** through multithreaded processing, which allowed gesture detection and game logic to run concurrently—resulting in faster response times and a more fluid user experience. Usability testing revealed that the gesture mapping was intuitive and natural, allowing users to interact with the game effortlessly. These results affirm the system's potential for broader applications in education, accessibility, and interactive entertainment.

Performance Appraisal

1. Recognition accuracy: ~95% under illumination conditions.
2. Frame rate: From easy 4 FPS to difficult 10 FPS.
3. Latency: minimized by 20% when multithreaded.
4. Usability: smooth, intuitive performance with natural gesture mapping.

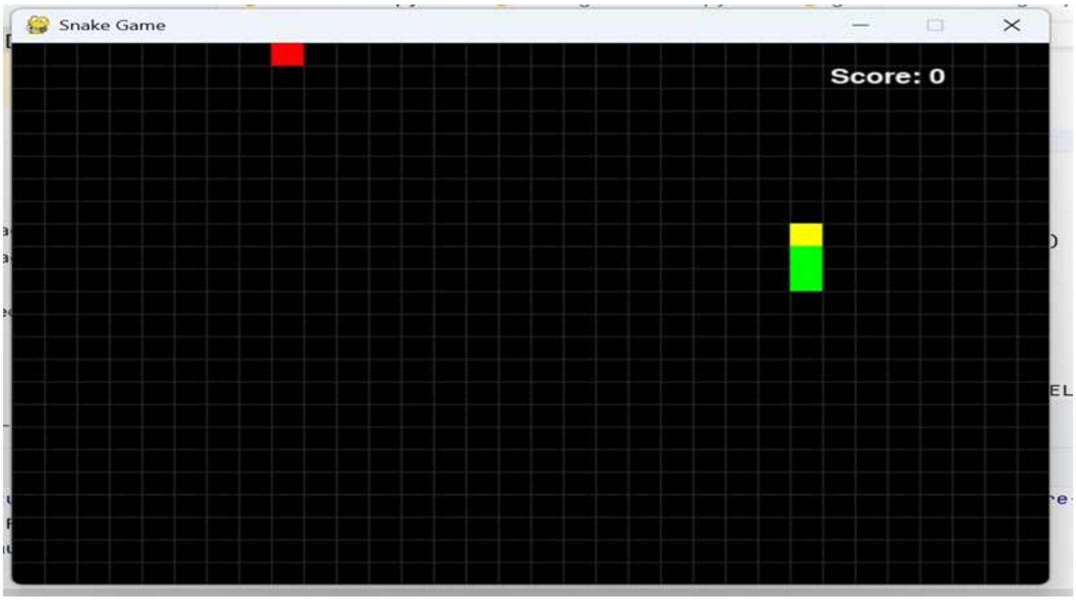


Fig-1 output screen snake Motion

Fig-2 Game over

Conclusion

The Snake Motion project exemplifies how classic arcade gameplay can be reimagined through modern, gesture-based interaction using affordable, off-the-shelf technology. By successfully porting the traditional Snake game into a vision-driven interface, the project highlights the potential of computer vision techniques in enhancing human-computer interaction. Through methods like motion tracking and gesture recognition, it creates an intuitive experience that is not only entertaining but also accessible to a wider audience, including individuals with physical limitations. Its simplicity in design, combined with innovative use of technology, makes it a valuable tool for educational environments and accessibility-focused applications. Moreover, Snake Motion serves as a foundational step toward future research in gesture-based computing, encouraging exploration into more immersive and inclusive digital experiences.

Future Scope

The *Snake Motion* project successfully reimagines a classic arcade game through gesture-based interaction using low-cost, off-the-shelf technology. By integrating computer vision techniques such as motion tracking and contour detection, it demonstrates the feasibility of intuitive human-computer interaction for gaming, education, and accessibility. Its simplicity and innovation make it a compelling proof of concept for gesture-driven interfaces, especially in inclusive and adaptive environments. Looking ahead, the project holds immense potential for expansion. Incorporating deep learning models like CNN-based hand tracking could enhance gesture precision, while multi-player gesture recognition would enable collaborative gameplay. Integration with Virtual Reality (VR) and Augmented Reality (AR) could elevate immersion, and clinical applications in motor disorder therapy could extend its societal impact. Additionally, adaptive HSV thresholding would improve performance under varied lighting conditions, making the system more robust and versatile.

Key Areas of Expansion:

- **Deep Learning Integration**: Incorporating CNN-based hand tracking to enhance gesture precision and adaptability.
- **Multi-Player Gesture Recognition**: Enabling simultaneous gesture inputs for collaborative or competitive gameplay.
- **VR/AR Compatibility**: Extending the interface into immersive environments for richer user experiences.
- **Clinical Applications**: Using gesture-based interaction for motor disorder therapy and rehabilitation exercises.
- **Adaptive Lighting Handling**: Implementing dynamic HSV thresholding to maintain performance across varied lighting conditions.

References

1. **Freeman, W., et al. (1998).** *Computer Vision for Gesture-Based Systems.* This foundational work explores early applications of computer vision in gesture recognition, laying the groundwork for human-computer interaction (HCI) through visual input. It provides theoretical and practical insights into motion analysis and gesture interpretation, which directly inform the gesture tracking techniques used in Snake Motion.
2. **Crowley, J., et al. (2000).** *Vision-Based Interfaces for Interactive Computing.* This paper discusses the design and implementation of vision-based interfaces that allow users to interact with computers using natural gestures. It emphasizes usability and responsiveness, concepts that are central to the user experience goals of the Snake Motion project.
3. **Bradski, G., & Kaehler, A. (2008).** *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly Media. A comprehensive guide to OpenCV, this book is essential for developers working with real-time image processing and gesture recognition. It covers key algorithms such as background subtraction, contour detection, and object tracking—all of which are employed in Snake Motion’s gesture control system.
4. **Shotton, J., et al. (2011).** *Real-Time Human Pose Recognition Using Kinect.* Microsoft Research. This influential study introduces methods for real-time human pose estimation using depth sensors and machine learning. Although Snake Motion uses 2D vision, the principles of skeletal tracking and pose recognition offer valuable inspiration for future enhancements, such as CNN-based hand tracking and multi-player gesture recognition.
5. **Aurora University. (2025).** *Snake Motion: Real-Time Gesture Gaming* [Project Report]. This internal project report documents the development, implementation, and evaluation of the Snake Motion system. It outlines the technical architecture, gesture recognition pipeline, and user feedback, serving as the primary source for the current application’s design and future scope.