Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 18s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Body Gesture Based Gaming, Math Solving and Mouse Controlling

Mr. Manish Mahajan[1], Dr. Lokendra Singh Songare[2]

[1]Research Scholar, Department of Computer Science and Engineering, Medi-Caps University, Indore, Madhya Pradesh, India, manish.mahajan@medicaps.ac.in
[2]Assistant Professor, Department of Computer Science and Engineering, Medi-Caps University, Indore, Madhya Pradesh, India, lokendra.singh@medicaps.ac.in

Cite this paper as: Mr. Manish Mahajan, Dr. Lokendra Singh Songare, et.al (2025) Body Gesture Based Gaming, Math Solving and Mouse Controlling. *Journal of Neonatal Surgery*, 14 (18s), 293-298.

ABSTRACT
The advancement of machine learning and computer vision technologies has surged, which enabled the idea behind the development of innovative gesture recognition systems that enhance human-computer interaction. "GenX: Move. Play. Solve." is an instigating project that leads to scope in gesture recognition technique to monitor & control the devices and applications through natural hand and body movements. This project introduces a versatile system that is capable of performing three distinct tasks: 'Mouse Controller, Game Controller, and Gesture Calculator'. Each module transforms user gestures into real-time, responsive actions, providing a seamless and intuitive user experience without the need for traditional input devices like keyboards or mouse. By applying modules for device control, gaming, and calculation, GenX aims to push the boundaries of gesture-based interaction and create new possibilities for accessibility, engagement, and productivity.

Keywords: Gesture Recognition, Human-Computer Interaction, Real-time Gesture Control, Gesture Controller, Accessibility Technology

1. INTRODUCTION

Recent developments in gesture detection technology have facilitated novel approaches to human-computer interaction. This technology provides a potential method for intuitive and hands-free operation of diverse computing devices, from conventional desktop configurations to immersive gaming environments.[1] This project, GenX, explores gesture-based control by offering a holistic system that integrates gesture detection for mouse manipulation, gaming interaction, and a novel gesture-based mathematical solver.[2,3]

The project revolves around leveraging the power of computer vision techniques to track hand movements and body poses in real-time using a standard webcam feed.[3] By employing the MediaPipe library for hand tracking and pose detection, coupled with PyAutoGUI for mouse emulation and game control,[4] we introduce a versatile system capable of interpreting user gestures and translating them into meaningful actions.

Our primary focus lies in providing users with a seamless and intuitive means of interacting with their computers. For mouse control, users can execute actions such as left click, right click, double click, and cursor movement through simple hand gestures captured by the webcam. This not only enhances productivity but also caters to individuals with mobility impairments by offering an accessible alternative to traditional mouse input methods. Furthermore, we extend the functionality of GenX to serve as a game controller for popular games like Subway Surfers.[4] We chose this because today's generation is heavily engaged in computer games, and sometimes even parents struggle to manage their gaming habits. This may lead to serious health issues in the long run. Hence, we implemented a game controller that is enjoyable for gamers while promoting healthier gameplay. By analysing the user's body pose and hand gestures, the in-game character can mirror the user's movements, enabling dynamic gameplay experiences. This integration of gesture-based control adds a layer of immersion and engagement, transforming gaming sessions into interactive adventures driven by natural gestures.[2,4-7]

Additionally, GenX incorporates a gesture-based math solver, allowing users to draw equations or mathematical problems with their fingers. Gemini (LLM) receives the drawn content when the user makes a fist, calculates it, and then displays the result to the user. This feature enhances the educational value of the project by providing an interactive method for solving mathematical problems.

Depending on their preferences and the task at hand, users can effortlessly transition between mouse control, game controller modes, and math solving due to our system's flexibility. GenX seeks to improve user experiences by offering a hands-free and intuitive interface that adjusts to various computing requirements, whether in applications, gaming, or mathematical problem-solving.

2. OBJECTIVES

Real-Time Gesture Recognition: Create a gesture recognition system that precisely identifies and monitors hand movements and body postures in real time utilizing a normal camera online.
Mouse Control Functionality: Develop features for mouse control, encompassing left click, right click, double click, cursor movement, and hold/release actions predicated on identified hand gestures.[8,9]
Game Controller Extension: Expand the project to operate as a game controller for widely played games such as Subway Surfers, enabling users to manipulate in-game character movements through intuitive hand gestures and body postures.[3,4]
Gesture-Based Math Calculator: Incorporate a gesture-based math calculator that enables users to do out mathematical operations by sketching expressions in space. Hand gestures are then translated to carry out computations and present the answers.[3,4]

3. REQUIREMENTS SPECIFICATION

Functional Requirements

The system must accurately detect and interpret hand gestures in real time, including movements such as swiping, tapping, and holding, in order to translate them into corresponding mouse or game controller actions. It should enable users to perform basic mouse functions, including cursor movement, left-click, right-click, double-click, and drag-and-drop, using hand gestures detected by the system. In addition to mouse control, the system must seamlessly integrate with compatible games, allowing users to control in-game actions such as character movement, jumping, crouching, and interacting with objects using hand gestures.
The system must reduce the amount of latency that occurs between the input of gestures and the actions that correspond to those gestures in order to offer a responsive and immersive user experience. This will ensure that users feel as though their interactions are instantaneous and natural.

Performance Requirements

In order to reduce the response time between gesture input and related actions, the system must be enough capable to recognize and interpret hand gestures in real-time. The gesture recognition reaction time should not be able to noticable by users, which should be less than 100 milliseconds(Ideally). While reducing false positives and false negatives, the system should exhibit high accuracy and precision in identifying and distinguishing between diverse hand motions. For the best reliability and user experience, the recognition accuracy should be higher than 95%. The system should be able to manage the performance when there are multiple concurrent users and input streams are presented to it. The system must use CPU, GPU, and memory resources optimally in order to reduce computational overhead and energy usage.
The system should able to perform stability and reliability over large usage periods, minimizing crashes, freezes, or unexpected behaviour like technical glithes. Before hardware implementations, the system must undergo thorough testing and validation to identify and reduce potential performance issues or software bugs.

Hardware Requirements

Computer system having moderate to high processing power and memory capacity is must to run the gesture recognition algorithms and game controller software effectively and efficiently.[10] ‘Recommended specifications include a multi core processor (e.g., Intel Core i5 or AMD Ryzen), at least 8 GB of RAM, and dedicated graphics processing unit (GPU) for enhanced performance. A high-definition webcam or camera capable of capturing video streams at a resolution of at least 720p is required for hand gesture recognition. The camera should have a wide field of view and sufficient frame rate to capture hand movements accurately.’
Audio output devices such as speakers or headphones are required for experiencing in-game audio signals, background music, and sound effects. Quality audio signals enhance the inerest and gaming experience to users.
Additional accessories like motion sensors, virtual reality (VR) headsets, or external microphones definitely enhance the gaming experience. Further, depending on the specific requirements of the project and user preferences from applications point of view.

Software Requirements

Obviously, there is a major concern related with the software requirements. The system's functionality depends on third-party software libraries, frameworks, or APIs for gesture recognition, system integration and development of games.[3,4] Project's stability and performance depend on compatibility issues or changes in the APIs’ parameters.

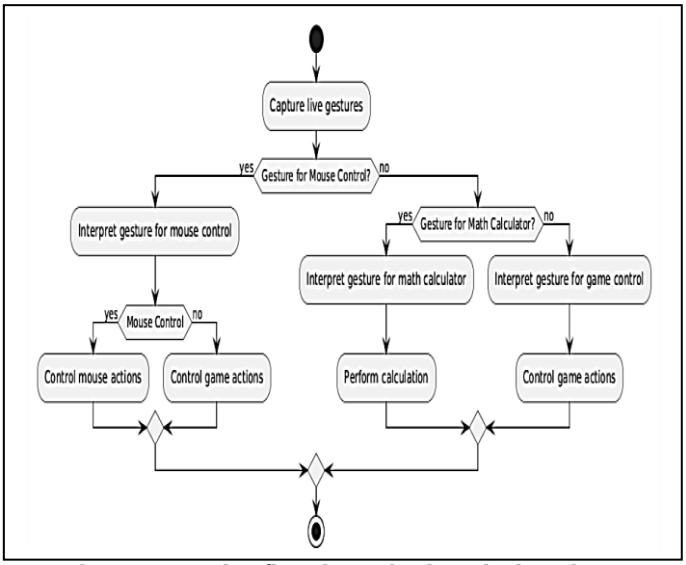
4. DESIGN

The system design for the gesture recognition mouse controller and game controller project encompasses several interconnected modules include to capture live gestures, recognize them, and translate them into corresponding actions & working together to enable real time gesture detection and control.[1-4]

Flow Chart

This flowchart illustrates the sequential steps involved in the project, from capturing input to processing gestures and controlling both the mouse and game character, along with considerations for user interface, optimization, documentation, and deployment.

1. Start: The process begins with capturing live gestures.
2. Decision Point: The flowchart checks whether the captured gesture is intended for mouse control.
3. If yes: If the captured gesture is for mouse control, the flowchart proceeds to interpret the gesture specifically for controlling the mouse.



4. Nested Decision: After interpreting the gesture, the flowchart checks whether the interpreted gesture indeed corresponds to mouse control.

• If yes: If the interpreted gesture is indeed for mouse control, the flowchart proceeds to control the mouse actions accordingly.

• If no: If the interpreted gesture does not correspond to mouse control, it is assumed to be for game control. The flowchart then proceeds to control game actions.

5. Else If (if the captured gesture is not for mouse control): If the gesture isn't for mouse control, it checks if it's intended for the math calculator. If yes, it interprets and performs the calculation.
6. Else (if the captured gesture is neither for mouse control nor math calculator): If the captured gesture is not intended for mouse control, the flowchart proceeds to interpret the gesture specifically for controlling the game.
7. Control Game Actions: After interpreting the gesture for game control, the flowchart proceeds to control game actions based on the interpreted gesture.
8. Stop: The process ends here.

5. IMPLEMENTATION AND TESTING

The implementation of project entitled 'GenX: Move. Play. Solve' involves the use of different languages, various integrated development environments (IDEs), tools, and techniques to achieve its defined objectives effectively.

By leveraging languages like 'Python Programming Language, Gemini with Google GenAI, OpenCV (Open-Source Computer Vision Library), MediaPipe, PyAutoGUI, Subprocess Module, Integrated Development Environments (IDEs) tools, Webcam or Camera, and technologies,' the project aims to implement an efficient gesture recognition mouse controller and game controller system, which offers an enhanced and interactive experience to users while interacting with computer applications and games through this system.

Testing procedures[11] are indispensable step of software development to ensure the reliability, functional accuracy, and performance of the implemented output. In the context of this project, some testing techniques and test plans can be employed that validate the system's behavior and identify potential issues and their effects on implemented output. Some key testing techniques that can be utilized for this project are:

- Unit Testing
- Integration Testing
- End-to-End Testing
- Regression Testing
- Performance Testing
- Usability Testing

6. RESULTS AND DISCUSSIONS

Mouse Controller UI:

1. Main Display:

- Live webcam feed occupying the entire screen.
- Hand landmarks overlaid on the webcam feed for gesture recognition.

2. Action Indicators:

- No specific action indicators are displayed as the focus is on gesture recognition for mouse control.

3. Status Bar:

- Instructions such as "Join both hands for game control" displayed.

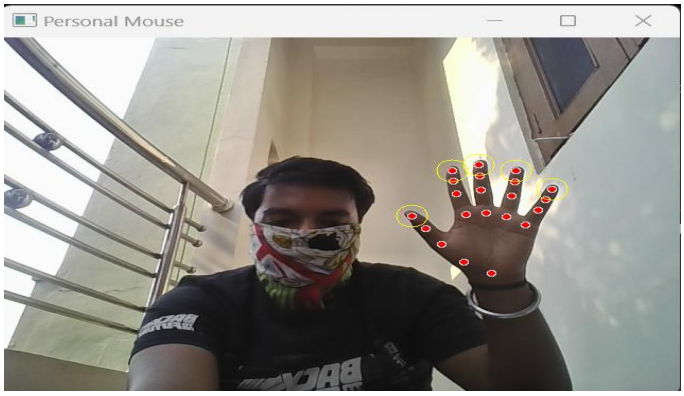


Figure 2: UI of Mouse Controller

Game Controller UI:

1. Main Display:

- Live webcam feed covering the majority of the screen. - Body landmarks overlaid on the webcam feed.
- Bottom left corner reserved for action display during gameplay (e.g., jumping, crouching).
- Instructions such as "Join both hands to start the game" displayed at the bottom left corner before starting the game.

2. Status Bar:

- X and Y axis.
- Top left corner displaying essential information:

• Frames per second (FPS) for real-time feedback.

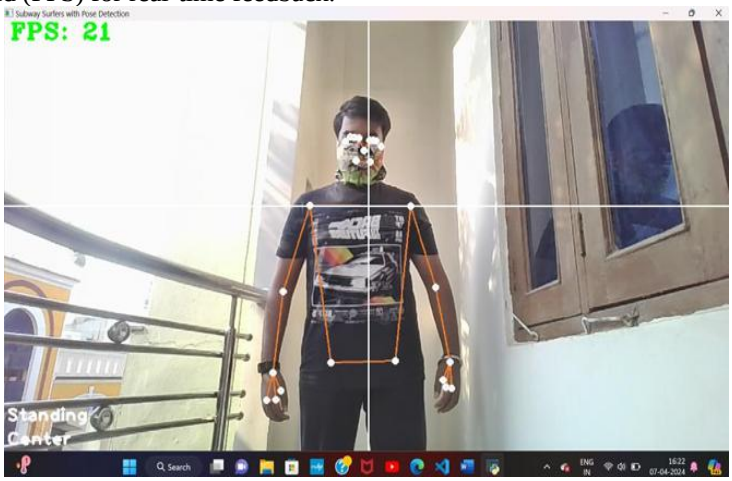


Figure 3: UI of Game Controller (After Start of the Game)

Math Calculator UI:

1. Main Display:

- Live Webcam Feed:

• The interface displays a live feed from the webcam, showing real-time hand movements.

• Hand Landmarks Overlay: Hand landmarks are overlaid on the webcam feed to indicate recognized gestures, which are essential for interpreting inputs for the calculator.

- Expression Display:

• Top-Center Area: The expression formed by hand gestures (e.g., “3 + 5”) is displayed in real-time. This allows users to verify their input as they gesture numbers and operators.

• Calculation Result: After making a specific gesture (e.g., joining both hands), the Gemini model processes the input and displays the calculated result in this same area.

2. Instruction and Status Panel:

- Bottom Display Panel:

- Instruction Prompts: Instructions like “Join both hands to calculate” guide users on performing specific gestures to trigger the calculator’s functions.

- Action Feedback: The UI provides feedback messages such as "Recognized number" or "Operator recognized," confirming that gestures have been correctly interpreted.

- Status Label: Displays messages like "Processing..." or "Calculation complete" to inform users about the status of their input

These UI designs aim to provide clear instructions and feedback to the user while maintaining a clean and unobtrusive interface. The mouse controller UI emphasizes the webcam feed and hand landmarks for gesture recognition, while the game controller UI integrates game action display seamlessly into the gameplay experience.

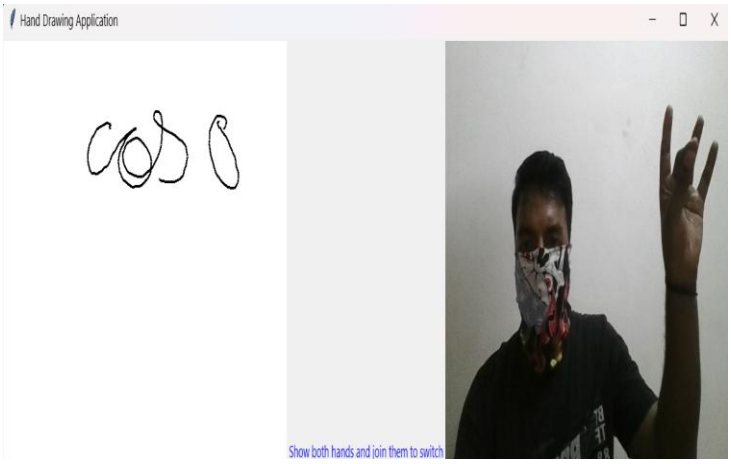


Figure 4: Math Calculator

7. CONCLUSION

The GenX project successfully demonstrated the potential of using hand gestures as a versatile input method for computer interactions, mathematical calculations, and gameplay, creating an immersive and engaging user experience. By eliminating the need for physical input devices, GenX offers an innovative, intuitive control mechanism that caters to a wide range of users, including those with mobility limitations or those looking for a more dynamic interaction experience.

8. ACKNOWLEDGMENTS

I would like to specially thank to GOD, my parents, my supervisor Dr Lokendra Singh Songare for their support and motivation, without their blessings and God’s grace this project would not be possible. They are gratefully acknowledged.

REFERENCES AND NOTES

1. Kumar, R., & Singh, A. “Gesture Recognition in Human-Computer Interaction: A Review.” Journal of Computer Science and Technology, Volume 35, Issue 4, pp. 809 825, 2020.
2. Zhang, Y., & Li, H. “Real-Time Gesture Recognition Using Machine Learning Techniques.” Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 123-129, 2021.
3. Szeliski, R. “Computer Vision: Algorithms and Applications.” Springer, Second edition, pp. 45-50, 2010.
4. 4. Kavin Chandar Arthanari Eswaran Akshat Prakash Srivastava Gayathri Mani “Hand Gesture Recognition for Human-Computer Interaction Using Computer Vision” Communications in Computer and Information Science, Deep Sciences for Computing and Communications (pp.77-90), March 2023.
5. Aashni Hariaa, Archanasri Subramaniana, Nivedhitha Asokkumara, Shristi Poddara , Jyothi S Nayaka “Hand Gesture Recognition for Human Computer
6. Interaction” 7th International Conference on Advances in Computing & Communications, ICACC-2017, 22-24 August 2017, Cochin, India.
7. Hongyu Zhou, Dongying Wang, Zhenrong Zhang “Research Progress of Human–Computer Interaction Technology Based on Gesture Recognition” Times New Roman, June 2023.
8. Rishabh Agrawal, Nikita Gupta “Real Time Hand Gesture Recognition for Human Computer Interaction” IEEE 6th International Conference on Advanced Computing, February 2016.

9. P. Kumar, S. S. Rautaray, and A. Agrawal, "Hand data glove: a new generation real-time mouse for Human-Computer In teraction," in Proceedings of the 2012 1st International Con ference on Recent Advances in Information Technology (RAIT), pp. 750–755, Dhanbad, India, March 2012.

10. N. Meghana, K. Sri Lakshmi, M. Naga Lakshmi Tejasree, K.Srujana,

11. N. Ashok "Gesture-based human-computer interaction" EPRA International Journal of Research and Development (IJRD) Volume: 8 | Issue: 10 | October 2023

12. Abeer Alnuaim, Mohammed Zakariah, Wesam Atef Hatamleh, Hussam Tarazi, Vikas Tripathi, Enoch Tetteh Amoatey "Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet" Lightweight Deep Learning Models for Resource Constrained Devices First published: 26 March 2022

13. Richard Szeliski. "Computer Vision: Algorithms and Applications" Comprehensive book covering computer vision fundamentals and Algorithms.