IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2025

DOI: 10.17148/IJARCCE.2025.14326

GesturaX

Swaraj Kanse[1], Raj Ghorpade[2], Prathamesh Tate[3], Mithun Mhatre[4]

Students, Department of Computer Technology, Bharati Vidyapeeth Institute of Technology, Navi Mumbai, Maharashtra, India[1,2,3]

HOD, Department of Computer Technology, Bharati Vidyapeeth Institute of Technology, Navi Mumbai, Maharashtra, India[4]

Abstract: GesturaX introduces a novel approach to touch-free computing by combining two interactive technologies—GestureSense and AirMouse—to redefine how users engage with digital systems. The GestureSense component employs advanced image processing techniques and machine learning algorithms, powered by the MediaPipe framework, to detect and interpret hand gestures in real time. This allows users to perform everyday tasks like moving cursors, clicking, scrolling, adjusting volume/brightness, and controlling media playback through intuitive hand motions.

The AirMouse module complements this by blending hardware innovation with motion sensing. Built around an ESP32-WROOM microcontroller and Bluetooth technology, it uses an MPU6050 inertial sensor for precise cursor control, alongside physical buttons and a rotary dial for clicks and scrolling. A compact 3.7V battery paired with a tp5100 charging unit and integrated safety systems ensures reliable, cord-free operation.

By merging visual gesture recognition with motion-based hardware controls, GesturaX offers a flexible solution for scenarios where traditional keyboards/mice fall short—from accessibility tools to gaming interfaces and smart home systems. This dual-modality design prioritizes both precision and ease of use, paving the way for more natural human-device interactions.

Keywords: Human-Computer Interaction, Gesture Recognition, Computer Vision, MediaPipe, ESP32, MPU6050, Bluetooth Connectivity, Touchless Interface, Sensor Fusion, Real-Time Control

I. INTRODUCTION

As technology reshapes how we live and work, the tools we use to interact with computers are ripe for reinvention. Keyboards and mice, while reliable, often limit users to desks or require physical contact—barriers in scenarios like presentations, gaming, or accessibility-focused applications. GesturaX reimagines this dynamic by blending two touch-free technologies: **GestureSense**, which interprets hand motions visually, and **AirMouse**, a motion- controlled hardware device. Together, they create a versatile, ergonomic alternative to traditional input systems.

GestureSense uses advanced image processing and machine learning to turn hand gestures into commands. By integrating MediaPipe's real-time tracking, it accurately recognizes movements as subtle as a finger tap or as dynamic as a sweeping motion. This lets users perform tasks like moving cursors, switching tabs, adjusting volume, or controlling media playback—all with intuitive hand motions. Beyond convenience, this system opens doors for users who rely on hands-free interfaces, such as those with mobility challenges or professionals managing smart home setups.

AirMouse takes a different approach, combining motion-sensing hardware with Bluetooth connectivity. Built around an ESP32-WROOM microcontroller and an MPU6050 motion sensor, it translates physical gestures—like tilting or rotating a handheld device—into precise cursor movements. Tactile buttons and a scroll wheel add familiarity, while a compact battery system (backed by safety safeguards) ensures hours of cordless use. Whether navigating a slideshow across the room or fine-tuning a 3D design, AirMouse bridges the gap between gesture-based control and tactile feedback.

By merging software-driven vision and hardware-based motion tracking, GesturaX offers a cohesive solution for modern computing needs. Its design prioritizes inclusivity—enabling users with disabilities to interact more freely— while also laying groundwork for immersive gaming and smart environments. This project isn't just about replacing keyboards and mice; it's about creating interactions that feel as natural as moving your hand or turning your wrist.

II. LITERATURE SURVEY

Human-computer interaction (HCI) has evolved dramatically in recent years, with touchless interfaces and gesture controls emerging as key frontiers. This section explores how gesture recognition and motion-sensing technologies have shaped the field, providing context for GesturaX's dual-modality design.

1. From Wearables to Vision: The Rise of Gesture Recognition

Early gesture-based systems relied on cumbersome wearables or colored markers to track hand movements—methods that felt unnatural and struggled with real-world reliability. The breakthrough came with computer vision, particularly deep learning frameworks that enabled markerless, real-time tracking. Innovations like Microsoft's Kinect demonstrated the potential of vision-based interaction, while modern tools like MediaPipe refined accuracy, detecting subtle hand landmarks without specialized hardware. These advancements inspired GesturaX's GestureSense module, which builds on these principles to translate gestures into commands like scrolling or brightness control, prioritizing both precision and ease of use.

2. Motion Sensors: The Hardware Side of Touchless Control

While cameras interpret gestures, sensor-based systems like AirMouse have taken a different path. Early versions used basic accelerometers to map motion to cursor movement, but accuracy was inconsistent. The integration of advanced sensors (like the MPU6050's inertial measurement unit) and efficient microcontrollers (such as the ESP32) transformed these devices into reliable tools. Modern AirMouse now blend motion tracking with tactile inputs—buttons for clicks, rotary dials for scrolling—creating a hybrid experience. This progress informs GesturaX's AirMouse, which pairs sensor-driven cursor control with familiar physical interactions, ensuring robustness even in low-light environments.

3. Bridging Software and Hardware: A Unified Approach

Recent HCI research emphasizes merging software algorithms with hardware innovation. For instance, combining vision- based gesture recognition with motion sensors can overcome individual limitations—like a camera's sensitivity to lighting or a sensor's drift over time. GesturaX adopts this philosophy, integrating MediaPipe's visual tracking with the ESP32's motion detection while addressing practical challenges like power efficiency. Lessons from prior studies guided key design choices, such as using compact batteries with smart charging (tp5100 modules) and prioritizing ergonomic shapes for comfortable, extended use.

4. Tradeoffs and Synergies

Vision and sensor systems each have strengths: cameras enable expressive, full-hand gestures, while motion sensors excel in consistency across environments. However, cameras struggle with poor lighting or obstructed views, and sensors may require occasional recalibration. GesturaX's hybrid model seeks to balance these tradeoffs. For example, a user could switch between GestureSense's detailed hand controls in a well-lit room and AirMouse's motion-based input in dim settings—a flexibility highlighted in studies as critical for real-world adoption.

The trajectory of HCI research—from early wearables to today's AI-driven vision systems—underscores the value of adaptable, multimodal interfaces. GesturaX builds on these foundations, merging the expressiveness of gesture recognition with the reliability of sensor hardware. By addressing gaps identified in prior work (like environmental adaptability and power efficiency), it aims to push touchless interaction into new domains, from assistive technologies to immersive gaming.

III. METHODOLOGY

GesturaX blends two touchless interaction modes—**GestureSense** (vision-based) and **AirMouse** (motion-sensing hardware)—into a unified system. Here's how we designed, built, and tested each component.

1. Designing the Dual-Mode System

The project's core idea is flexibility: users can switch between gesture-based control (for expressive tasks) and a handheld AirMouse (for precision). We developed both modules separately, then merged them into a single platform. Think of it as giving users a "gesture remote" and a "motion mouse" in one toolkit.

2. GestureSense: Teaching Cameras to Understand Hands

Setting Up the Camera and Software

- A standard webcam (set to 640×480 resolution) captures live video. The feed is flipped (like a mirror) and colour-adjusted for compatibility with gesture-tracking tools.

- Key Tools:
- *MediaPipe*: Detects 21 hand landmarks (fingertips, joints, wrist) in real time.
- *OpenCV*: Processes video frames and overlays visual cues.
- *PyAutoGUI*: Converts gestures into clicks, scrolls, or keyboard shortcuts.

How Gestures Become Commands

- Finger Logic: The system checks which fingers are raised. For example:
- Thumb position (left vs. right) helps distinguish gestures like "scroll up" or "volume down."
- A raised pinky triggers scrolling; a thumb-and-index combo switches tab.
- Movement Mapping: Hand sweeps adjust volume or brightness. The system maps hand positions to screen coordinates, ensuring a wave to the left lowers volume, while a wave to the right raises it.

Making It Smooth and User-Friendly

- *Anti-Jitter*: A smoothing algorithm averages cursor positions to prevent shaky movements.
- *Calibration*: Users “train” the system once to align hand movements with their screen size.
- *Feedback*: A live preview (built with PyQt) shows bounding boxes and text (e.g., “Volume Up”) so users know their gestures are detected.

3. AirMouse: Building a Motion-Sensing Remote

Hardware Setup

- Core Components:
- *ESP32-WROOM*: Acts as the brain, handling Bluetooth communication.
- *MPU6050 Sensor*: Tracks tilts and movements (like a Wii remote).
- *Tactile Buttons*: For clicks; a rotary dial (like a mouse wheel) handles scrolling.
- *Battery System*: A rechargeable 1000mAh battery with safety circuits for all-day use.

Firmware: Turning Motion into Cursor Control

- The ESP32 reads motion data from the MPU6050 and converts it into cursor movement. For instance:
- Tilting the device left/right moves the cursor horizontally.
- A button press sends a click signal via Bluetooth.
- *Noise Reduction*: Filters clean up shaky sensor data, ensuring smooth tracking.

Testing and Refinement

- We fine-tuned sensitivity so small hand tremors don’t jitter the cursor.
- Users tested the device in varied settings (e.g., dim rooms, crowded desks) to ensure reliability.

4. Integration: Making Both Modules Work Together Seamless Switching

Users toggle between modes via a menu. For example:

- Use GestureSense during a presentation (hands-free slide control).
- Switch to AirMouse for detailed tasks like photo editing.

Calibration Across Environments

- *Lighting Tests*: GestureSense was adjusted to work in both bright and low-light conditions.
- *Motion Calibration*: AirMouse’s sensitivity was tweaked for different desk setups (e.g., sitting vs. standing).

User Feedback Loops

- Visual cues (on-screen overlays) and tactile responses (button clicks/LED blinks) help users troubleshoot. For example:
- If GestureSense doesn’t detect a hand, the GUI prompts users to reposition.
- AirMouse’s LED flashes red when the battery is low.

5. Why This Approach Works

By combining vision and hardware, GesturaX overcomes the limitations of each method:

- **GestureSense** excels in expressive tasks (e.g., adjusting settings mid-game).
- **AirMouse** thrives in precision scenarios (e.g., editing spreadsheets).
- Together, they create a system that adapts to users’ needs—whether they’re navigating a smart home or need hands-free accessibility.

DOI: 10.17148/IJARCCE.2025.14326

6. Future Directions: Building on GesturaX

While GesturaX achieves its core goals, future work could refine and expand its capabilities:

- **Teaching GestureSense New Tricks**: Adding more gestures (e.g., sign language support) and refining accuracy in cluttered or dim environments.
- **Smarter Sensors**: Integrating depth cameras (like Intel RealSense) to track 3D hand movements, reducing occlusion issues.
- **Making AirMouse Feel Alive**: Adding haptic feedback (e.g., subtle vibrations) to confirm clicks or scrolls, mimicking the “click” of a physical mouse.
- **Adapting to Users**: Personalizing gesture sensitivity or motion thresholds based on individual preferences or accessibility needs.

These upgrades could position GesturaX for applications in virtual reality, healthcare (e.g., sterile environments), or industrial control systems.

7. Conclusion: A Step Toward Intuitive Interaction

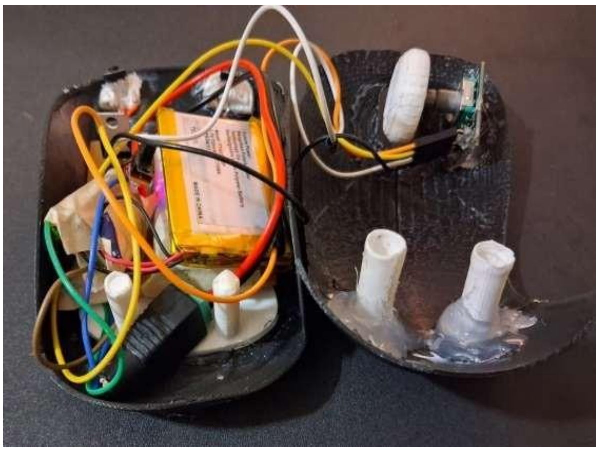
GesturaX bridges the gap between gesture-based and hardware-driven touchless control. By combining the expressiveness of computer vision (GestureSense) with the reliability of motion sensors (AirMouse), the project demonstrates how hybrid systems can overcome the limitations of single-mode interfaces.

Key Strengths:

- *User-Centric Design*: From anti-jitter algorithms to ergonomic hardware, every component prioritizes ease of use.
- *Real-World Testing*: Iterative refinements based on diverse environments ensures robustness.
- *Accessibility*: The system’s flexibility empowers users with mobility challenges or those working in hands-free settings.

Looking ahead, GesturaX’s modular architecture allows for easy integration of emerging technologies, from AI-driven gesture prediction to ultra-low-power sensors. By focusing on both technical precision and human needs, the project lays a foundation for interfaces that feel less like tools and more like natural extensions of our actions—whether in smart homes, gaming, or collaborative workspaces.

IV. RESULT



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2025

DOI: 10.17148/IJARCCE.2025.14326

Welcome to GestureSense

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.

Click the 'GestureSense' button to start or stop gesture recognition.

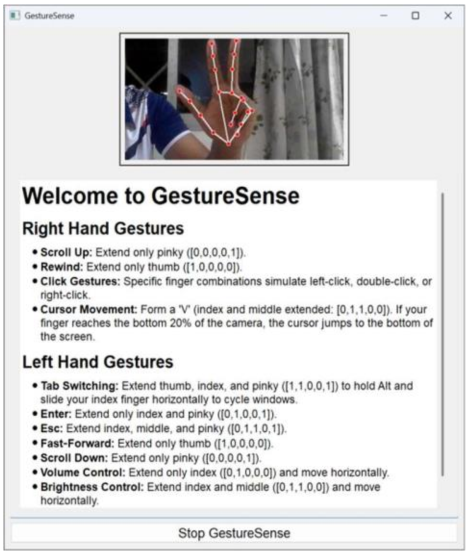
Welcome to GestureSense

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.



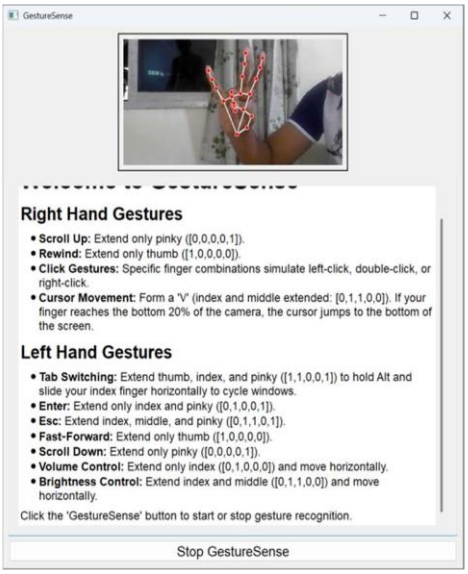
Welcome to GestureSense

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.



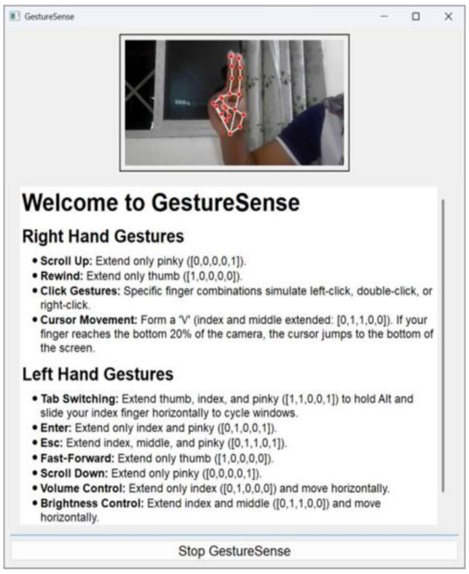
Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.

Click the 'GestureSense' button to start or stop gesture recognition.



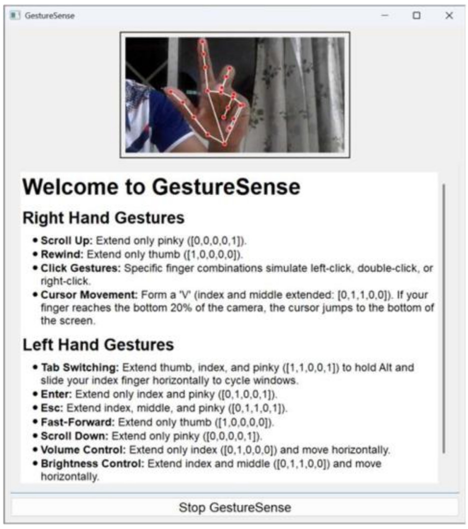
Welcome to GestureSense

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.



Welcome to GestureSense

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.

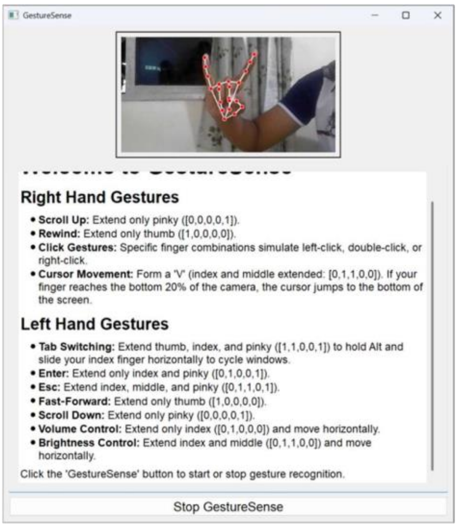
Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.

Click the 'GestureSense' button to start or stop gesture recognition.



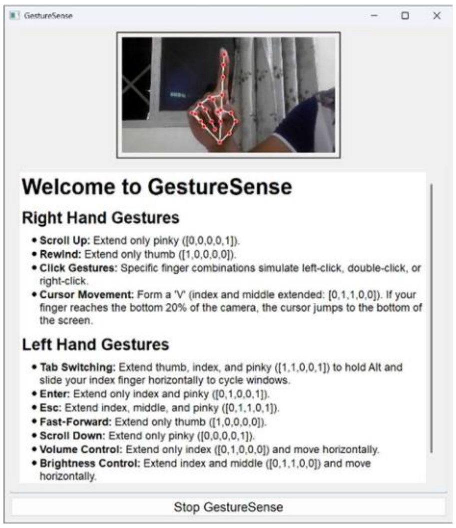
Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.

Click the 'GestureSense' button to start or stop gesture recognition.



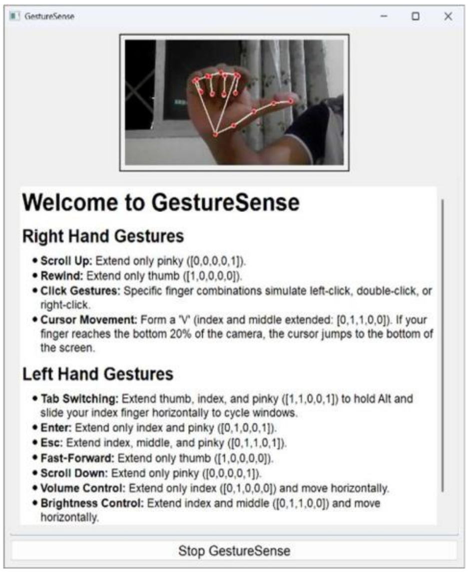
Welcome to GestureSense

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.



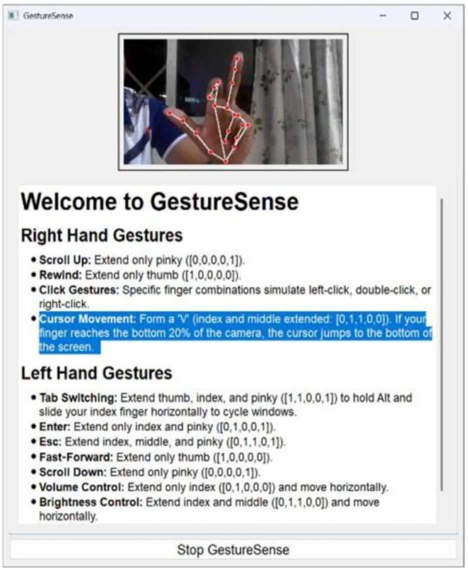
Welcome to GestureSense

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.



Welcome to GestureSense

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.

DOI: 10.17148/IJARCCE.2025.14326

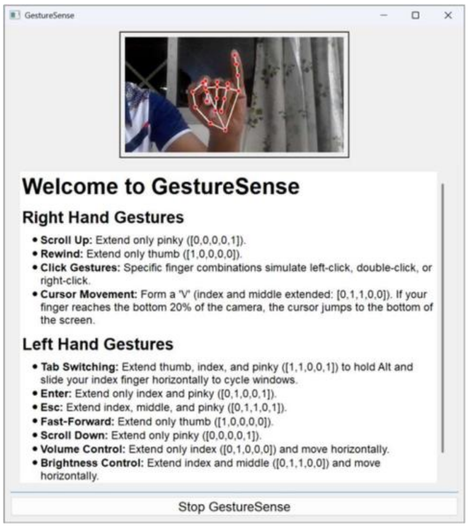
Welcome to GestureSense

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.



Welcome to GestureSense

Right Hand Gestures

- **Scroll Up:** Extend only pinky ([0,0,0,0,1]).
- **Rewind:** Extend only thumb ([1,0,0,0,0]).
- **Click Gestures:** Specific finger combinations simulate left-click, double-click, or right-click.
- **Cursor Movement:** Form a 'V' (index and middle extended: [0,1,1,0,0]). If your finger reaches the bottom 20% of the camera, the cursor jumps to the bottom of the screen.

Left Hand Gestures

- **Tab Switching:** Extend thumb, index, and pinky ([1,1,0,0,1]) to hold Alt and slide your index finger horizontally to cycle windows.
- **Enter:** Extend only index and pinky ([0,1,0,0,1]).
- **Esc:** Extend index, middle, and pinky ([0,1,1,0,1]).
- **Fast-Forward:** Extend only thumb ([1,0,0,0,0]).
- **Scroll Down:** Extend only pinky ([0,0,0,0,1]).
- **Volume Control:** Extend only index ([0,1,0,0,0]) and move horizontally.
- **Brightness Control:** Extend index and middle ([0,1,1,0,0]) and move horizontally.

REFERENCES

[1]. Sachin Singh. “AI Virtual Mouse Using Hand Gesture Recognition.” Medium, 14 June 2023.
[2]. Ranjith GC, Saritha Shetty. “Implementing a Real Time Virtual Mouse System Using Computer Vision.” International Research Journal of Modernization in Engineering Technology and Science, May 2023.
[3]. Sen, Abir; Mishra, Tapas Kumar; Dash, Ratnakar. “Deep Learning Based Hand Gesture Recognition System and Design of a Human-Machine Interface.” arXiv:2207.03112, July 2022.
[4]. Xu, Pei. “A Real-time Hand Gesture Recognition and Human-Computer Interaction System.” arXiv:1704.07296, April 2017.
[5]. Google. “MediaPipe: A Framework for Building Perception Pipelines.” arXiv:1906.08172, 2019. [6]. “Project Digits.” Wikipedia.
[7]. “Gesture Recognition.” Wikipedia.
[8]. Espressif Systems. “ESP32-WROOM Module Documentation.” [9]. InvenSense. “MPU6050 Datasheet.”