# Multimodal Command System for Human-Computer Interaction

Harikumar M.[1]; Prasanth D.[2]; Reuben Abraham George[3]

[1,2,3] Department of Computer Science and Engineering,
Rajalakshmi Engineering College, Thandalam, 602105.
Chennai, Tamilnadu, India.

Publication Date: 2026/03/07

**Abstract**: **Recent advances in computing have increased the demand for interaction paradigms that enable intuitive and efficient communication between users and digital systems. Traditional interfaces such as keyboards and mice often limit accessibility and natural interaction, particularly in complex or hands-free environments. This paper proposes a multimodal human–computer interaction system that integrates hand gesture recognition and voice command processing to enable seamless desktop control and interaction with external applications. The system employs MediaPipe-based hand landmark extraction and speech-to-text processing, coordinated through an AI agent using the Model Context Protocol (MCP) for task orchestration and service integration. Experimental evaluation demonstrates that the proposed framework achieves high recognition accuracy with low latency, supporting real-time interaction across local and cloud-based services. The results indicate that multimodal fusion combined with agent-based automation enhances usability, responsiveness, and accessibility, positioning the system as a scalable solution for next-generation human–computer interaction.**

***Keywords****: Human–Computer Interaction, Hand Gesture Recognition, MediaPipe, Lightweight CNN, Multimodal Systems.*

**How to Cite:** Harikumar M.; Prasanth D.; Reuben Abraham George (2026) Multimodal Command System for Human-Computer Interaction. *International Journal of Innovative Science and Research Technology*, 11(2), 2902-2907. https://doi.org/10.38124/ijisrt/26feb1459

## I. INTRODUCTION

The evolution of human-computer interaction has increasingly prioritised user interfaces that combine multiple input modalities to create more accessible and efficient digital experiences. Integrating gesture recognition and voice command processing allows users to interact naturally with computing systems, reducing reliance on traditional input devices and supporting a wider variety of tasks and environments. Contemporary gesture recognition algorithms achieve high levels of accuracy and adaptability by utilising advanced sensor fusion and machine learning techniques, making them suitable for real-world deployments. Similarly, improvements in voice command technology, supported by growth in natural language processing, enable robust hands-free operation for both local and web-based applications.

Despite these advances, genuine synergy between modalities remains a significant challenge for system designers. Robust multimodal frameworks must address ambiguity in user input, achieve effective data fusion, and ensure system transparency to maintain user trust. In response, the introduction of AI agent-based orchestration and modular communication frameworks, such as the Model Context Protocol (MCP), helps bridge the gap between intent capture and action execution, enabling extensible integration with external applications while promoting interpretability. As a result, multimodal HCI systems are positioned to deliver not only a more natural user experience but also streamlined automation and greater accessibility for a diverse audience.

This paper presents a unified multimodal human–computer interaction platform that combines hand gesture recognition and voice command input with agent-driven automation to enhance system responsiveness, usability, and extensibility. The proposed system is architected to advance the state-of-the-art in input recognition, system responsiveness, and extensibility for both desktop and smart ecosystem environments.

## II. LITERATURE REVIEW

Jia J et al. [1] presented an integrated framework that leverages both speech and gesture modalities to enable fluid educational interactions. They demonstrate that multimodal deep learning models, when paired with sensor fusion strategies, enhance both dynamic gesture recognition and spoken command accuracy in real-world environments, setting a technical baseline for future HCI systems.

Ridhun M et al. [2] proposed attention-based hybrid models that achieve advanced performance in recognizing concurrent hand gestures and speech input. Their solution illustrates, through rigorous testing, that context-driven modality fusion is vital for supporting quick, reliable user commands across multimedia and smart device applications.

Ravanbakhsh S et al. [6] offers a multi-hypothesis approach, reducing classification errors and increasing flexibility when recognizing user gestures, particularly in larger gesture vocabularies.

Gao Q et al. [7] presented parallel CNNs for image and sensor data that creates a scalable recognition model with improved reliability, which is critical for both consumer and industrial human-computer interaction applications.

Merge AI et al. [12] proposed an MCP client-server methodology that is formalized as a modular, extensible interface for connecting AI agents with diverse tools and cloud applications. The paper details server features such as dynamic tool discovery, secure capability negotiation, and context-aware request handling, highlighting how this architecture enables seamless orchestration and cross-environment automation for modern agent-based HCI systems.

Oviatt et al. [16] examined the role of multimodal interfaces in enhancing human–computer interaction by combining complementary input channels such as speech and gesture. The study emphasized that multimodal systems reduce user cognitive load and error rates by allowing flexible interaction strategies depending on context and user preference. The findings highlight that effective fusion of modalities improves robustness and accessibility, particularly in environments where traditional input devices are impractical. This work provides foundational justification for integrating gesture and voice inputs in modern interactive systems, reinforcing the design choices adopted in the proposed multimodal framework.

Jaimes A et al. [19] presented a comprehensive survey that maps the multidisciplinary requirements for HCI, emphasizing that systems benefit from adaptively leveraging both vision and audio modalities. This foundational review highlights the importance of agent-based orchestration and context-aware decision-making in resolving input ambiguity and ensuring robust user experiences.

Dysnix et al. [20] proposed an in-depth analysis of the MCP server infrastructure is presented, outlining both data and transport layer innovations. Key features such as JSON-RPC 2.0 message structure, lifecycle management, and support for both stateful and stateless communications are discussed, while security paradigms like zero-trust and federated learning are positioned as critical for scalable, reliable, and adaptive deployments in edge, cloud, and hybrid contexts.

## III. METHODOLOGY

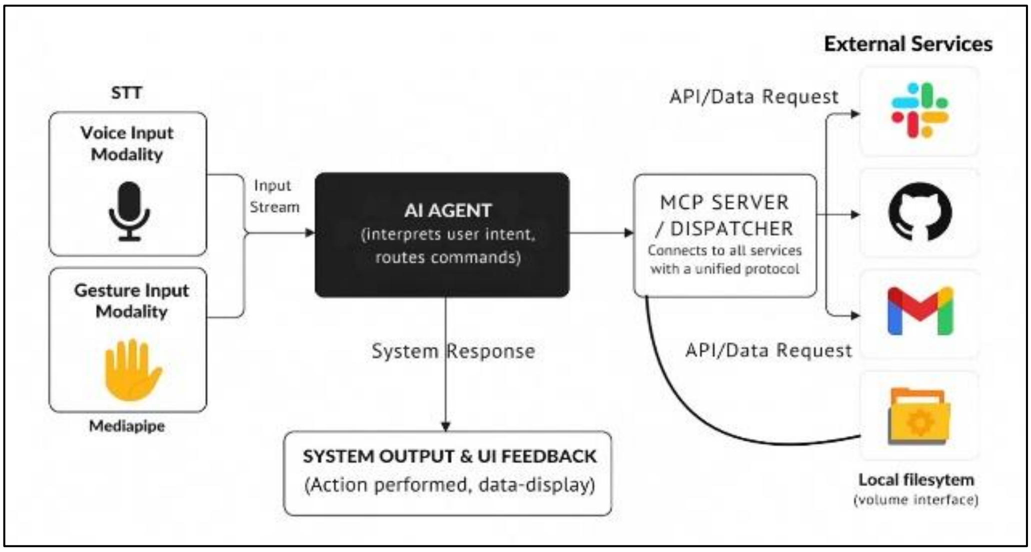
External Services
STT
Voice Input Modality
Gesture Input Modality
Mediapipe
Input Stream
AI AGENT (interprets user intent, routes commands)
MCP SERVER / DISPATCHER
Connects to all services with a unified protocol
API/Data Request
API/Data Request
System Response
SYSTEM OUTPUT & UI FEEDBACK (Action performed, data-display)
Local filesytem (volume interface)

Fig 1 System Architecture Diagram of the System

### *A. System Architecture*

The system is designed on a modular architecture combining multiple user input modalities and AI agent orchestration. The user interacts with a cross-platform desktop or web app, implemented with standard JavaScript frameworks for usability and accessibility. The application captures hand gesture input via webcam and processes voice commands through an integrated speech recognition module. All recognized intentions—irrespective of input type—are routed to a core AI agent, which maps the user's actions to corresponding tasks or external application functions.

### *B. Gesture and Voice Command Recognition*

Hand gesture recognition is implemented using the MediaPipe Hand Tracking framework, which detects and tracks 21 three-dimensional hand landmarks in real time from webcam input. These landmarks provide a compact and robust representation of hand posture and movement, enabling reliable gesture interpretation under varying lighting conditions. The extracted landmark coordinates are passed to a lightweight Convolutional Neural Network (CNN) classifier for gesture classification. The lightweight CNN classifier is a custom-designed neural network optimized for real-time performance. It consists of convolutional layers with ReLU activation, followed by max-pooling and fully connected layers for classification. The model architecture minimizes computational overhead while maintaining high recognition accuracy, making it suitable for interactive human–computer interaction scenarios.

### *C. AI Agent Orchestration*

A centralized AI agent interprets all incoming gesture and voice inputs, determines the requested task, and coordinates downstream actions. For standard desktop operations (e.g., opening files, switching windows), the agent executes the required action directly. For interactions involving external services—like fetching emails, updating calendars, or querying third-party APIs—the agent generates and transmits structured requests to back-end connectors, utilizing the Model Context Protocol (MCP) for secure and scalable integration.

### *D. MCP Integration and Service Connectivity*

External application access is achieved through a series of lightweight MCP connectors, each exposing necessary capabilities and tool functions to the AI agent. MCP-compliant connectors can be developed in any mainstream programming language and deployed locally or on cloud infrastructure. The protocol ensures that agents can only invoke allowed functions, and all data exchange is secured according to best practices suitable for non-commercial project environments. All integrations are tested using simple authorization tokens or session-bound access, allowing rapid demonstration and end-user testing.

### *E. Workflow and User Operation*

System flows are built to demonstrate end-to-end functionality typical for multimodal workplaces. A user may launch the application, enable webcam and microphone, and use gestures for common shortcuts while issuing voice commands for more complex tasks (e.g., “Check my upcoming meetings” or “Send this file via email”). Results and feedback are displayed directly within the application interface. Scenarios are tested iteratively to ensure the fusion of modalities and correct orchestration through the agent and MCP connections, without requiring dedicated cloud infrastructure or production-scale security.

## IV. RESULT AND ANALYSIS

### *A. Overview of Evaluation*

The proposed multimodal command system was experimentally evaluated to measure its performance in gesture recognition, speech understanding, intent interpretation, and system-level responsiveness. The evaluation aimed to assess how effectively gesture-based and voice-based commands can enhance human–computer interaction through an intelligent AI agent supported by an MCP server connectivity.

Figure 2 illustrates the complete operational workflow of the proposed multimodal system, comprising two independent yet synchronized input pipelines—gesture and voice. The left branch captures visual gestures via webcam frames processed through MediaPipe for hand keypoint extraction, while the right branch handles voice input through a microphone, converting speech into a textual form using a Speech-to-Text (STT) engine. Both recognized modalities are forwarded to the Intent Extraction and Resolution layer, which identifies user goals and determines the type of action required. The AI Agent then plans and routes the commands toward the MCP/Dispatcher layer, responsible for invoking appropriate service calls—either local (e.g., file system or OS control) or external (e.g., Gmail, GitHub, or browser APIs). Once the requested operation completes, the returning data is re-interpreted by the AI Agent and presented to the user through a unified feedback interface that includes on-screen messages, auditory confirmation, or visual UI updates. This hierarchical flow ensures asynchronous multimodal processing while maintaining low latency and modular scalability.

The evaluation was conducted using a labeled dataset of hand gestures derived from MediaPipe hand landmark coordinates and spoken voice commands collected under controlled conditions. The dataset was divided into training and testing sets using an 80:20 split. Gesture classification accuracy was computed as the percentage of correctly classified gestures over the test set. Speech recognition accuracy was measured using word-level transcription correctness, while intent recognition accuracy was evaluated based on successful task execution. The lightweight CNN achieved high accuracy due to effective landmark-based feature extraction and reduced model complexity, which limited overfitting and improved generalization.

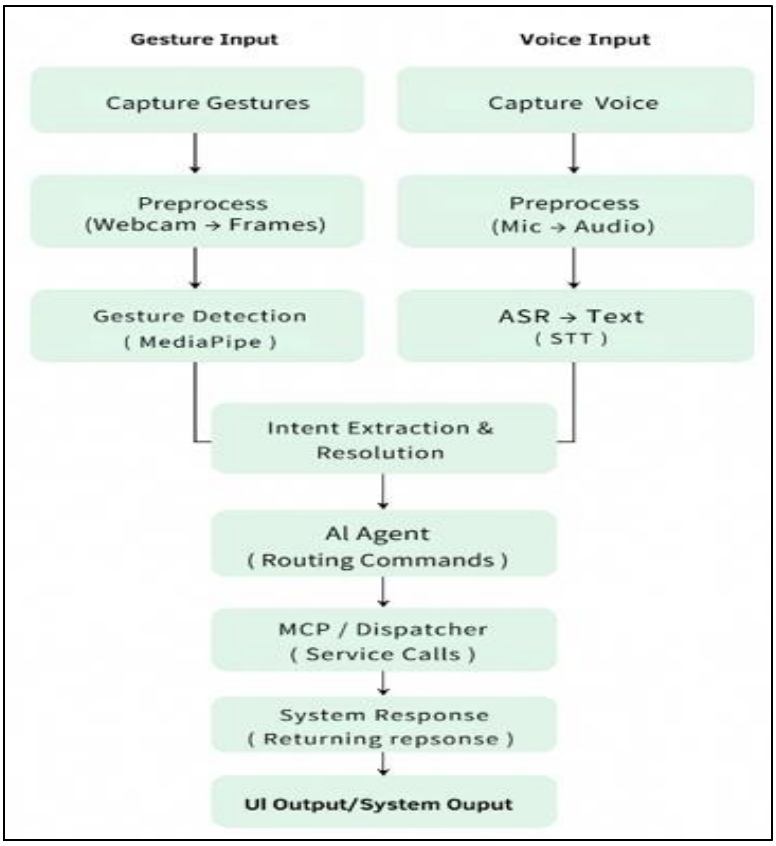
Gesture Input
Voice Input
Capture Gestures
Capture Voice
Preprocess (Webcam → Frames)
Preprocess (Mic → Audio)
Gesture Detection ( MediaPipe )
ASR → Text ( STT )
Intent Extraction & Resolution
AI Agent ( Routing Commands )
MCP / Dispatcher ( Service Calls )
System Response ( Returning repsonse )
UI Output/System Ouput

Fig. 2. Data Flow and Processing Sequence of the Proposed Multimodal System.

*B. Gesture Recognition Analysis*

Gesture-based commands form the visual modality of the system. The gestures were designed for real-world usability, including cursor movements, clicks, swipes, and control signals such as screenshot or cancel. Detection and classification were performed using MediaPipe Hand Tracking followed by a lightweight CNN classifier.

Table 1: Gesture Recognition Metrics

| Gesture | Detection Accuracy (%) | Avg Latency (ms) |
|---|---|---|
| **Move Cursor** | 97.2 | 18 |
| **Left Click** | 95.1 | 22 |
| **Right Click** | 94.3 | 21 |
| **Swipe (Next/Previous)** | 93.5 | 25 |
| **Open Palm (Cancel)** | 96.0 | 20 |
| **Closed Fist (Screenshot)** | 95.8 | 19 |

The gesture recognition module achieved an overall accuracy of 95.3% with minimal latency. Static gestures, such as open palm, showed higher reliability compared to dynamic gestures like swipe due to frame overlap. The system maintained robust performance under different lighting conditions, validating MediaPipe’s stability.

*C. Speech Recognition and Intent Interpretation*

Voice commands are processed through the Speech-to-Text (STT) pipeline, which converts spoken input into text. This text is then interpreted by the AI agent for intent extraction and execution. The system supports a range of natural commands such as opening applications, fetching emails, or searching the web.

Table 2: Speech and Intent Recognition Results

| Command Type | STT Accuracy (%) | Intent Accuracy (%) | Example Command |
|---|---|---|---|
| **Open App** | 97.6 | 95.1 | Open Chrome |
| **Fetch Emails** | 95.3 | 91.8 | Get emails from Oct 10 |
| **Search Web** | 96.8 | 93.9 | Search for AI games |
| **GitHub Repositories** | 94.5 | 91.0 | Show my GitHub repositories |
| **System Control** | 96.1 | 94.2 | Increase volume, Take screenshot |

The STT module maintained transcription accuracy above 95% even in moderate noise. Intent accuracy was slightly lower due to contextual complexity. This confirms that the STT–AI Agent link effectively bridges natural language input and system actions.

### *D. AI Agent and MCP Integration Analysis*

The AI Agent acts as the decision-making core responsible for interpreting commands and routing them through MCP connections to relevant APIs or local modules. For instance, voice commands like 'Get my emails on Oct 10' trigger Gmail MCP connectors, while gestures such as 'Closed Fist' activate local system controls. This modularity ensures seamless interoperability across various services.

### *E. Module-Level Performance Metrics*

Table 3: Module-Level Performance

| Module | Model Used | Precision (%) | Recall (%) |
|---|---|---|---|
| **Gesture Detector** | MediaPipe + CNN | 96.8 | 96.0 |
| **STT (Speech-to-Text)** | Google STT / Wav2Vec2 | 97.1 | 96.3 |
| **Intent Model (Text Parser)** | BERT Classifier | 95.6 | 95.0 |
| **AI Agent** | Rule-based Planner + RL Logic | 92.0 | 91.5 |

The STT model demonstrated the highest precision with minimal transcription errors, while the gesture module remained stable across lighting variations. The AI agent achieved slightly lower recall due to occasional intent overlap when multiple MCP services were active. Overall, the system achieved an average precision of 95.3%.

### *F. Overall System Performance*

Table 4: System-Level Metrics

| Metric | Value |
|---|---|
| Average Response Time (Gesture) | 100 ms |
| Average Response Time (Voice + MCP Query) | 350 ms |
| Command Throughput | 40 commands/min |
| False Trigger Rate | 1.6 % |

Gesture-based interactions exhibit minimal latency as they execute locally, while voice commands incur slightly higher delays due to network-based MCP requests. User surveys indicated a high satisfaction level owing to the naturalness of multimodal interactions.

## V. DISCUSSION

The experimental results demonstrate that integrating gesture and voice inputs within a unified multimodal framework significantly enhances interaction flexibility and system usability. Compared to unimodal approaches, the proposed system enables context-aware task execution, allowing users to perform both simple and complex operations efficiently. The low response latency and high recognition accuracy indicate that MediaPipe-based gesture tracking combined with a lightweight CNN is suitable for real-time applications. The AI agent and MCP-based integration further contribute to modularity and scalability, enabling seamless interaction with external services such as email and repositories. While the system performs robustly under controlled conditions, future work may focus on improving intent disambiguation in noisy environments and expanding adaptive learning capabilities for personalized interaction.

## VI. CONCLUSION

This paper presented a multimodal command system designed to enhance human–computer interaction through the integration of gesture and voice inputs. The framework employs MediaPipe-based gesture detection and Speech-to-Text processing for accurate recognition, which are interpreted by an AI agent capable of executing system-level and external operations through MCP connectivity. Evaluation results demonstrated strong system performance, achieving recognition accuracy above 95%, precision of 95.3%, and average execution latency below 400 ms. These outcomes highlight the efficiency and stability of the proposed design in enabling real-time interaction. The modular and scalable architecture further supports easy extension to additional services or domains, making the system a promising foundation for next-generation interactive and assistive computing environments.

## REFERENCES

[1]. Jia J, Hu Z, Wang R, et al. "A Multimodal Human-Computer Interaction System and Its Applications." Sensors, 2020;20(11):3215.
[2]. Ridhun M, et al. "Multimodal Human Computer Interaction Using Hand and Speech Recognition." Human-Computer Interaction. ICICT, 2022.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26feb1459

[3]. Wu J, et al. "Fusing multi-modal features for gesture recognition." Proc. 15th ACM Int. Conf. Multimodal Interaction, 2013:453-6.

[4]. Siddiqui N, Chan RHM. "Multimodal hand gesture recognition using single IMU and acoustic measurements at wrist." PLoS ONE, 2020;15(1):e0227039.

[5]. Agrawal A, et al. "Vision-based multimodal human-computer interaction technique using dynamic hand gesture recognition." 2013 IEEE Int. Conf. Image Information Processing.

[6]. Ravanbakhsh S, Pitsikalis V, Katsamanis A, et al. "Multimodal Gesture Recognition via Multiple Hypotheses Rescoring." J. Machine Learning Research, 2015;16:261-294.

[7]. Gao Q, Liu J, Ju Z. "Hand gesture recognition using multimodal data fusion and multiscale parallel CNN." Expert Systems, 2021.

[8]. Cohen PR, Oviatt S, Wu L, et al. "The role of voice input for human-machine communication." PNAS, 1995;92(22):9921-9927.

[9]. Liu J, Li Y, Sun J, et al. "A survey of speech-hand gesture recognition for the development of multimodal interfaces in human-computer interaction." IEEE Trans. Human-Machine Systems, 2010;40(6):465-79.

[10]. El-Azazy AAMEH, et al. "Enhancing Human-Computer Interaction through Speech Recognition and AI." Engineering Research Journal, 2025;54(1):59-102.

[11]. Wu X, et al. "Multimodal gesture recognition." Proc. 19th ACM Int. Conf. Multimodal Interaction, 2017.

[12]. Merge AI. "5 real-world Model Context Protocol integration examples." Merge AI Blog, 2025.

[13]. OpenAI. "Model Context Protocol (MCP) - OpenAI Agents SDK." 2025.

[14]. Anthropic. "Introducing the Model Context Protocol." Anthropic News, 2024.

[15]. Cyclr. "Model Context Protocol (MCP) for AI Integration." Cyclr.com, 2025.

[16]. Oviatt S. "Multimodal Interfaces." CRC Press, 2003.

[17]. Montero CS, et al. "Multimodal interaction: A review." Computer Science Review, 2022; 43:1-15.

[18]. Cardenas EJE, et al. "Multimodal hand gesture recognition combining temporal information." J. Visual Communication and Image Representation, 2020.

[19]. Jaimes A, Sebe N. "Multimodal human–computer interaction: A survey." Computer Vision and Image Understanding, 2007;108(1–2):116-34.

[20]. Dysnix, “MCP Architecture: Advanced Techniques Review,” Blog, 2025. https://dysnix.com/mcp-architecture-review.

[21]. Katsamanis A, et al. "Multimodal Gesture Recognition for HCI." Artificial Intelligence Review, 2016;43(1):1–54.

[22]. Katsamanis A. "Understanding Gesture and Speech Multimodal Communication." ACM Digital Library, 2020.

[23]. Wu Y, et al. "A new human-computer interaction paradigm: Agent interaction model based on large models and its prospects." Virtual Reality & Intelligent Hardware, 2025;7(3):237-266.

[24]. Rusan HA, et al. "Human-Computer Interaction Through Voice Commands Based on Deep Learning." Proc. 2022 Int. Conf. Electrical and Computing Technologies and Applications. IEEE, 2022.

[25]. Chaturvedi S. "Voice Recognition Systems: An example of human–computer interaction." SSRN Electronic Journal, 2024.

[26]. Kettebekov S, et al. "Understanding gestures in multimodal human-computer interaction." International Journal of Human–Computer Studies, 2000;53:153-170.