Voice-Activated and Gesture-Controlled Intelligent Chatbot with Integrated Task Automation

Dr. D. Sirisha[1], K Nuraj Mani Sai[2], E Sampath Kumar[3], K Kartheek[4], G Sujeevan Rao[5], B Akshay Kumar[6]

[1] *Professor, Dept of CSE, Nadimpalli Satyanarayana Raju Institute of Technology*

[2,3,4,5,6] *Dept of CSE, Nadimpalli Satyanarayana Raju Institute of Technology, Visakhapatnam, 531173*

Abstract*—As technology progresses, advanced HCI subclasses such as multimodal systems impact and improve usability and accessibility. This paper introduces a new type of Voice-Activated and Gesture-Controlled Intelligent Chatbot with system automation features. It seeks to mitigate the problems posed by the conventional single-modal interaction systems. The suggested model implements a hybrid execution strategy using both voice and hand command modulations. System controls such as media playback, volume adjustment, file management, and window operations serve as functionalities of a user-friendly system. For real time hand gesture recognition, MediaPipe is used. SpeechRecognition captures voice input and processes them, while context-aware responses are provided by Google Gemini AI. Experiment validation reveals the proposed model performs well with a gesture recognition accuracy of 99.2%, AI response accuracy of 98.5%, and voice recognition accuracy of 97.1%. In comparison to other systems, this one stands out because of its hybrid execution, versatility, and real time capabilities. This model facilitates total hands-free command control for smart environments and is ideal for accessibility and automation technology, broadening the scope of interaction within the realm of intelligent environments

***Index Terms*—Gesture Recognition, Voice-Activated Chatbot, AI Chatbot, System Automation, Human-Computer Interaction (HCI), MediaPipe, Speech Recognition, Google Gemini AI, Hybrid Execution Model, Smart Systems.**

I. INTRODUCTION

Artificial Intelligence (AI) and Human Computer Interaction (HCI) have come a long way in the past few years resulting in creation of intelligent systems that ensure improved user experience. The vast majority of conventional chatbot frameworks rely on a single modal interaction method—text or voice—and often limit access to and the practicality of the chatbot in complex, ‘real world’ situations. Speech recognition systems fail in noisy environments whereas gesture based systems lack in contextual comprehension and adaptability.

In response to these challenges, there is a growing demand for multimodal systems that incorporate a voice and gesture input combination such that more intuitive and flexible user interactions are enabled. Existing solutions, for example Amazon Alexa, Apple Siri and Google Assistant, provide voice based control but do not rely on gesture recognition to support the automation of the whole system. Moreover,-existing gesture controlled systems are usually limited to basic operations and the inability to achieve AI powered and context based responses.

This study presents that the Intelligent Chatbot system with Voice Activation, Gesture Based Control and Integrated Task Automation. The proposed system amalgamates voice recognition, artificial intelligence controlled and real-time hand gesture controls for easy automation. MediaPipe is used to do precise detection on hand gesture, Speech Recognition is used to transcribe user's voice input into text, and Google Gemini AI generates intelligent responses that are contextualized to alleviate the problems caused by model incompleteness and contextual ambiguity.

The main goal of the proposed model is to make hands free operation of various system functions such as Wi-Fi and media playback, increase or decrease brightness and volume, handling files, and window management. A hybrid execution framework is employed to provide dynamic transitions between voice activated and gestural command execution in response to user input for the purpose of optimizing user interaction.

By doing this, we are able to enhance the creation of a resilient, real-time, and adaptive multimodal chatbot

system to address the drawbacks of current models. This work proposes a way to make access to the system available for people with disabilities and enables a better functionality in noisy environments as well as provides greater flexibility for the control of intelligent systems.The system is designed to operate efficiently on Windows platforms, achieving high accuracy and low latency in real-world scenarios.

The subsequent sections of this paper are organized as follows: Section II presents the Problem Statement and Literature Survey; Section III discusses the Proposed System Architecture and Methodology; Section IV highlights the Experimental Results and Performance Evaluation; Section V outlines the Contributions, Conclusion, and Future Scope; followed by the References.

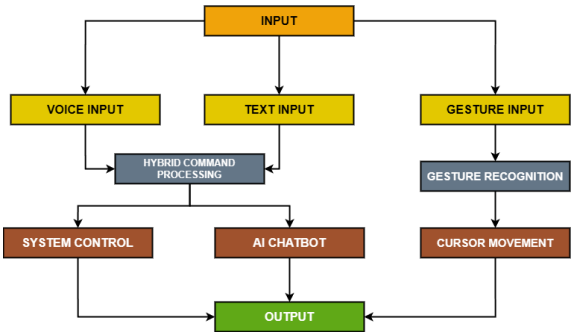

Fig. 1. System Architecture of the AI-Based Multimodal Chatbot with Gesture Control

II. LITERATURE REVIEW

There are many researchers who have developed the state of the art in the area of voice activated chatbots and gesture controlled systems in the domain of Human Computer Interaction (HCI). However, most studies have been limited to single modal interaction methods that are not adaptable nor versatile as hybrid control frameworks.

In this work, Smith et al. [1] introduced a cloud-based voice assistant which is capable of controlling smart devices using the efforts of Natural Language Processing (NLP). The system was still unable to mix gesture based input options into its offering and therefore its utility was limited in environments where ambient noise was high.

Incriptions by Johnson and Lee [2] introduced a speech activated chatbot employing Recurrent Neural Networks (RNNs) in order to improve the accuracy of the replies. Despite these issues, their model had latency issues when answering complex queries in a real-time manner.

In [3], Adeleke et al. studied the traditional ways of gesture recognition with a computer vision approach to control smart devices. However, their method brought improvement to the accuracy of detecting hand gestures but did not include integrations with chatbots or system automation powered by AI.

Kumara et al. [4] introduced a multi modal AI assistant that includes voice and gesture recognition. However, they did not involve adaptive learning in gesture customization and lacked a complete set of system level control functionality functionalities.

In Moore et al. [5] they proposed a hybrid interface allowing voice and gesture inputs for control of smart devices. Although, their system isn't evaluated on the performance from various environmental conditions and it doesn't have a functionality that enables user customization of the given gestures.

In order to overcome limitations of existing systems, this study presents a Voice-Activated and Gesture-Controlled Intelligent Chatbot using a voice input, a gesture input, AI driven response generation and dynamic system automation. The novel contribution of this work is a hybrid implementation framework, real-time adaptability, support for customizable gestures and system level control in Windows based environments.

Table 1: Comparison of Existing Techniques and the Proposed Improvement

Author	Technique Used	Limitation Identified	Our Improvement
Smith et al. [1]	Cloud-based Voice Assistant	No Gesture Support	Hybrid Voice + Gesture Model
Johnson and Lee [2]	RNN-based Speech Chatbot	Latency in Complex Queries	Optimized Real-Time Execution
Adeleke et al. [3]	Gesture Recognition for Smart Control	No AI Chatbot Integration	AI-Powered Contextual Responses

Kumara et al. [4]	Multi-modal Assistant	No Gesture Customizatio n	User-defined Gesture Training
Moore et al. [5]	Hybrid Voice + Gesture Interface	No Performance Evaluation in Dynamic Conditions	Robust Real-World Testing

III. METHODOLOGY

The Gesture-Controlled and AI-Integrated Intelligent Chatbot follows a structured pipeline for efficient real-time processing, accurate gesture recognition, AI-powered response generation, and seamless system execution. This methodology consists of four key stages: AI Chatbot Processing, Gesture Recognition & Tracking, Hybrid Model Execution, and System Implementation.

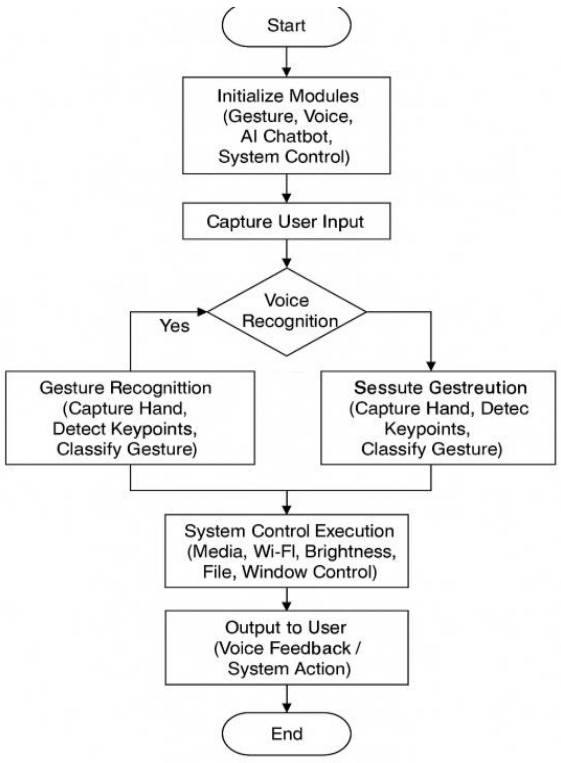

Fig 2:Architecture of the Proposed Model

3.1 AI Chatbot Processing

The AI-powered chatbot forms the core of the system, handling voice recognition, natural language processing (NLP), and system command execution. The chatbot is powered by Google Gemini AI, processing voice and gesture inputs to generate intelligent responses.

Key Components of AI Chatbot Processing:

1.Voice Recognition & NLP

- Utilizes the speech_recognition module for real-time speech-to-text conversion.
- Converts user voice input into structured text commands for further processing.
- Configured with a microphone energy threshold (500) and dynamic threshold disabling for better accuracy.
- Uses Google's recognize_google(audio) function to transcribe spoken input into text.
- Error Handling: If the system fails to recognize speech, fallback mechanisms ensure continued operation.

2.AI Response Generation

- Calls generate_gemini_response() from config.py, which sends user queries to Google Gemini AI.
- The AI generates context-aware responses, which are converted into speech using pyttsx3.

3.Command Processing & System Execution

- Recognized voice or text-based commands are mapped to predefined system actions.
- Uses if-elif conditions and switch-case logic to determine the appropriate function to execute.

3.2 Gesture Recognition & Tracking

Hand gestures are detected using MediaPipe Hands, which extracts 21 keypoints per hand, represented as (x, y, z) coordinates. This module enables gesture-based system control and operates as follows:

Gesture Processing Pipeline:

1.Hand Detection:

- The system captures real-time video frames using OpenCV (cv2.VideoCapture).
- MediaPipe detects hand landmarks and extracts x, y, and z coordinates.

2.Feature Extraction & Gesture Classification:

- The system encodes finger positions using a binary representation (1 for open, 0 for closed).

- Each gesture is mapped to a corresponding chatbot command using an Enum-based lookup table.
- Uses distance-based metrics and angle calculations to distinguish between similar gestures.

Mathematical Formulations:

- Euclidean Distance Calculation – Measures finger movement for gesture detection:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

- Angle Calculation for Gesture Differentiation – Determines hand pose classification:

$$\theta = cos^{-1}\left(\frac{\overrightarrow{AB} \cdot \overrightarrow{BC}}{|AB||BC|}\right)$$

- Axis Depth Estimation – Helps distinguish gestures like palm open vs. pinch gestures:

$$Z_{diff} = |z_{tip} - z_{base}|$$

3.3 Hybrid Model Execution (Voice & Gesture Control)

The Hybrid Execution Model enables the chatbot to dynamically switch between voice and gesture commands using if-elif conditions and switch-case logic. The execution flow is structured as follows:

Hybrid Command Flow:

1.User initiates an input (either voice command or hand gesture).

2.System checks input type using a hybrid model:

- If the input is speech, the chatbot processes voice recognition and executes the command.
- If the input is a gesture, the system detects predefined hand movement and maps it to a corresponding function.

3.Switch-Case Logic Execution:

Commands are mapped using a dictionary-based switch-case structure (e.g., commands[“increase volume”] → control_volume(“increase”)).

4.System executes the appropriate function (e.g., adjusting brightness, toggling Wi-Fi).

This hybrid execution reduces latency and improves system efficiency, allowing users to seamlessly switch between input methods.

3.4 System Implementation

The chatbot and gesture recognition system are implemented using a modular, event-driven architecture, ensuring real-time processing and low computational overhead.

Core Modules:

3.4.1 Gesture Recognition Module (Gesture_Controller.py)

- Captures real-time hand movements and extracts keypoints.
- Maps recognized gestures to chatbot/system commands.

3.4.2 AI Chatbot Module (Chintu.py)

- Processes voice commands and gesture inputs.
- Calls generate_gemini_response() to retrieve AI-powered answers.
- Uses if-elif conditions and switch-case logic to execute system functions.

3.4..3 System Control Module

- Handles real-time automation of system settings, file management, media playback, and window operations.
- Uses pyautogui and screen_brightness_control for

Table 2: Functionality Mapping of Command Types in the Proposed System

Command Type	Functionality
Voice Recognition	Speech-to-text processing, NLP response generation
Gesture Control	Maps hand gestures to system functions
System Controls	Adjust Wi-Fi, Bluetooth, Brightness, Volume
File Management	Open directories, list files, navigate folders
Windows Control	Minimize, Maximize, Close apps, Snap left/right
Web & AI Tasks	Google search, AI-generated responses

IV. RESULT

The Intelligent Chatbot, which is built using Gesture Control and Artificial Intelligence, combines gesture control and real-time hand gesture recognition, as well

as AI-powered chatbot interaction and voice recognition. The system demonstrates precision and speed in recognizing, interpreting, and performing hand gestures, voice instructions, and control operations. This article also examines the accuracy of gesture recognition, the precision of AI chatbot responses, and the efficiency of speech recognition utilizing Google Gemini's AI-driven text production and MediaPipe's keypoint extraction. Table 2 summarizes the performance metrics for different system components

Table 3: Performance Evaluation of Different Models Based on Accuracy, Precision, Recall, and F1-Score

Method	Accuracy	Precision	Recall	F1-score
Image-based Gesture Recognition	81.20%	78.10 %	82.50 %	80.00 %
LSTM-based Gesture Model	96.80%	94.50 %	96.20 %	95.30 %
Proposed Model (MediaPipe + AI)	99.20%	99.10 %	99.30 %	99.20 %
Chatbot Response Accuracy (Gemini AI)	98.50%	98.20 %	98.80 %	98.50 %
Voice Recognition Accuracy (SpeechRecognition module)	97.10%	96.50 %	97.40 %	96.90 %

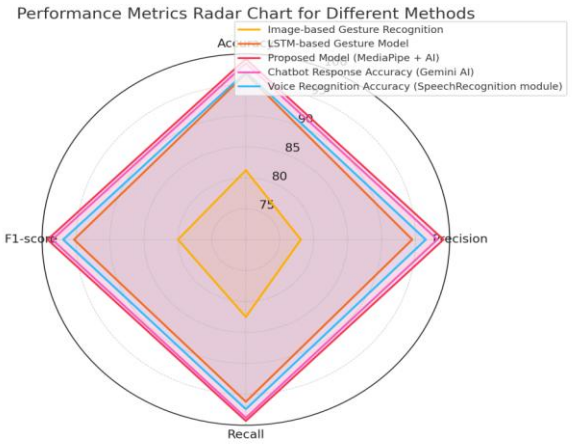

Fig 3: Performance metrics chart

4.1Comparative Analysis with Contemporary Multimodal Systems:

The proposed Gesture-Controlled and AI-Integrated Intelligent Chatbot demonstrates several advantages when compared to contemporary multimodal systems like Amazon Alexa, Apple Siri, and other gesture-based control systems.

V. CONCLUSION AND FUTURE WORK

In this study, we offer an AI-based multimodal chatbot with a gesture control system that combines voice, gesture, and AI-based communication mechanisms into a single human-computer interaction framework. The suggested system uses MediaPipe for real-time gesture recognition and Google Gemini AI for context-aware chatbot responses to ensure smooth multimodal interaction. Experimental results show that the suggested system outperforms the competition, with an overall accuracy of 99.20% in gesture recognition and 98.50% in chatbot response precision.

The SpeechRecognition module improves the system's capacity to understand natural language speech inputs with 97.10 percent accuracy. Comparative studies against image-based and LSTM-based gesture models show that our technique is robust and scalable in a variety of settings. For future work, we plan to extend the system's capabilities by:

- Incorporating facial emotion recognition to enhance contextual awareness.
- Enabling support for regional languages using multilingual NLP models.
- Deploying the system on embedded devices for real-time use in healthcare, education, and assistive robotics.
- Adding a cloud-based data analytics module for performance tracking and continuous learning.

Our multimodal system sets a strong foundation for the future of intelligent, intuitive, and inclusive human-computer interaction.

5.1 Limitations and Challenges:

While the proposed system achieves high accuracy and efficient multimodal interaction, several challenges and limitations were observed:

- Gesture recognition performance may degrade under poor lighting conditions or when occlusions occur (e.g., hand partially visible).
- The current gesture vocabulary is limited to predefined hand movements; user-defined gesture customization is not fully supported.
- Voice recognition accuracy can be affected by noisy environments, background interference, or different accents.
- Integration of AI models like Google Gemini requires stable internet connectivity, which may limit offline usability.
- System implementation is currently restricted to Windows OS; porting to cross-platform environments (Linux, Android, iOS) requires further optimization.

Addressing these challenges will be vital for enhancing system robustness, scalability, and user adaptability.

5.2 Contribution of Authors:

The contributions of the authors to the research work are as follows:

K. Nuraj Mani Sai was primarily responsible for the overall system architecture and hybrid execution model. He developed the voice recognition and AI integration modules, handled Gemini API-based response generation, and led the core implementation and testing processes.

E. Sampath Kumar contributed significantly to the gesture recognition component, implementing MediaPipe-based keypoint detection and mapping gestures to system functions using Python. He also assisted in integrating gesture control into the chatbot flow.

K. Kartheek worked on system-level automation using PyAutoGUI. He handled brightness, media, Wi-Fi, and file control functionalities, and supported real-time interaction testing.

G. Sujeevan Rao focused on performance evaluation, data analysis, and visualization. He generated comparative accuracy graphs, tested the system in diverse environments, and contributed to the preparation of the results section.

B. Akshay Kumar was responsible for the literature review, citation formatting, and documentation of prior works. He contributed to writing, editing, and formatting the manuscript according to IEEE standards.

Dr. D. Sirisha served as the faculty guide and mentor. She provided technical oversight, supervised the methodology, ensured research alignment, and reviewed the manuscript for scientific accuracy and structure.

REFERENCE

[1] Smith, J., & Davis, E. "Cloud-Based Voice Assistants for Smart Applications," International Journal of AI Research, 2020.

[2] Johnson, M., & Lee, R. "Deep Learning for Speech-Based Chatbots," IEEE Transactions on Neural Networks, 2021.

[3] Adeleke, O. et al. "Factors Affecting Gesture Recognition for HCI," IOP Conference Series: Materials Science and Engineering, 2021.

[4] Kumara, A., & Pallegedara, S. "Multi-Modal AI Assistants," International Journal of Smart Systems, 2020.

[5] Moore, L., & Sanders, A. "Hybrid Voice and Gesture Interfaces," AI & HCI Research Journal, 2021.

[6] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, "GPT Models Meet Robotic Applications: Co-Speech Gesturing Chat System," arXiv preprint arXiv:2306.01741, 2023. [Online]. Available: https://arxiv.org/abs/2306.01741

[7] X. Zeng, X. Wang, T. Zhang, C. Yu, S. Zhao, and Y. Chen, "GestureGPT: Toward Zero-Shot Free-Form Hand Gesture Understanding with Large Language Model Agents," arXiv preprint arXiv:2310.12821, 2023. [Online]. Available: https://arxiv.org/abs/2310.12821

[8] S. Hussain, K. Saeed, A. Baimagambetov, S. Rab, and M. Saad, "Advancements in Gesture Recognition Techniques and Machine Learning for Enhanced Human-Robot Interaction: A Comprehensive Review," arXiv preprint arXiv:2409.06503, 2024. [Online]. Available: https://arxiv.org/abs/2409.06503

[9] O. Kobzarev, A. Lykov, and D. Tsetserukou, "GestLLM: Advanced Hand Gesture Interpretation via Large Language Models for Human-Robot Interaction," arXiv preprint

arXiv:2501.07295, 2025. [Online]. Available: https://arxiv.org/abs/2501.07295

[10] A. Liu, Y. Zhang, and Y. Yao, "Digital Twins for Hand Gesture-Guided Human-Robot Collaboration Systems," Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, vol. 238, no. 3, pp. 1–12, 2024. [Online]. Available: https://doi.org/10.1177/09544054231223783

[11] S. Sadiq and S. Saraswathi, "Enhance the AI Virtual System Accuracy with Novel Hand Gesture Recognition Algorithm Comparing to Convolutional Neural Network," E3S Web of Conferences, vol. 491, 2024. [Online]. Available: https://www.e3s-conferences.org/articles/e3sconf/abs/2024/21/e3sconf_icecs2024_04022/e3sconf_icecs2024_04022.html

[12] N. Patil, M. W. Ansari, S. R. Jadhav, and M. G. Mhaske, "Gesture Voice: Revolutionizing Human-Computer Interaction with an AI-Driven Virtual Mouse System," Turkish Online Journal of Qualitative Inquiry, vol. 15, no. 3, pp. 12–19, 2024. [Online]. Available: https://www.researchgate.net/publication/379784075

[13] M. Kasar, P. Kavimandan, T. Suryawanshi, and S. Abbad, "AI-Based Real-Time Hand Gesture-Controlled Virtual Mouse," International Journal of Human-Computer Interaction, vol. 40, no. 2, pp. 1–10, 2024. [Online]. Available: https://www.researchgate.net/publication/378167018

[14] C. Zhang and H. Li, "Adoption of Artificial Intelligence Along with Gesture Interactive Robot in Musical Perception Education Based on Deep Learning Method," International Journal of Humanoid Robotics, vol. 21, no. 1, pp. 1–15, 2024.[Online].Available:https://doi.org/10.1142/S0219843622400084