iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** I **Month of publication:** January 2026

DOI: https://doi.org/10.22214/ijraset.2026.77040

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

Emotion Based AI Assistant with Dynamic System Control

Kamatchi T P[1], Suganya J[2], Pranav A[3], Surya Kesavan[4], Vivek D[5], Nishanth B[6]
Department of Computer Engineering, PSG Polytechnic College, Coimbatore, India

Abstract: This project presents the design and development of an Emotion-Based AI Assistant with Dynamic System Control and Hand Gesture Interaction, implemented entirely on a standard laptop environment using Python. Unlike conventional AI assistants that rely solely on voice commands, the proposed system integrates voice recognition, hand gesture recognition, contextual natural language understanding, real-time operating system control, and emotion-aware interaction into a unified and intelligent desktop platform. User interaction is enabled through speech input and visual hand gestures, allowing seamless and contactless control of the system. Voice commands are interpreted using the Gemini 2.5 Flash AI model, while the system further enhances human-computer interaction through an emotion detection module implemented using OpenCV and the DeepFace library. Continuous emotion re-evaluation ensures real-time responsiveness and personalized interaction, such as triggering adaptive actions like playing calming background music during negative emotional states.
Keywords: Affective Computing, Gemini 2.5 Flash, MediaPipe, DeepFace, Human-Computer Interaction, Python, Gesture Control, Voice Automation.

I. INTRODUCTION

Artificial Intelligence (AI) has rapidly evolved into a cornerstone technology shaping the future of human-computer interaction (HCI). Traditional interaction methods such as keyboards, mice, and touchscreens, although effective, require physical effort and direct user involvement.

With the advancement of computer vision and machine learning, interaction paradigms are gradually shifting toward more natural, contactless, and intelligent systems that can understand human behaviour, gestures, and emotions. This evolution aims to create "Affective Computing" environments where systems do not merely execute commands but also understand the psychological context and emotional state of the user.While voice-based AI assistants like Google Assistant, Apple Siri, and Amazon Alexa represent a significant milestone, they are largely optimized for mobile devices and cloud-based ecosystems. Their functionality on personal computers remains limited, particularly regarding direct operating system (OS) control, gesture-based interaction, and emotional awareness. Furthermore, a heavy dependence on cloud infrastructure raises critical concerns regarding privacy, latency, and functionality in offline environments. Existing solutions often lack the stability or real-time responsiveness required for practical desktop deployment, leading to fragmented workflows and impersonal, mechanical communication.

II. OBJECTIVES

The objectives are:

1) To design and implement a real-time hand gesture recognition system using computer vision techniques.
2) To develop a multi-mode interaction framework supporting system control, media control, and presentation management.
3) To integrate emotion detection through facial expression analysis and adapt system responses accordingly.
4) To implement secure mode switching using a two-hand gesture validation mechanism to prevent accidental changes.

III. WORKING PRINCIPLE

The system begins with the acquisition of live video input through a webcam, which processes the user's hand movements and facial expressions. For hand gesture recognition, landmark detection algorithms identify 21 key points, such as finger joints and tips. Simultaneously, emotion recognition algorithms evaluate features like eye movement and mouth shape to identify states such as happiness, stress, or neutrality.

Once a gesture or emotion is validated, the system maps the interpreted input to corresponding OS-level actions like adjusting brightness or launching applications.

IV. MODULES

The modules in the project are:

1) Voice Recognition and Command Processing: Module 1 serves as the auditory cortex of the assistant, responsible for bridging the gap between acoustic sound waves and machine-understandable intent. Unlike basic voice-command tools that rely on rigid keyword matching, this module utilizes Generative AI to handle the fluidity and ambiguity of natural human speech. The technical workflow begins with passive wake-word monitoring, where a lightweight listener constantly monitors the audio stream for specific frequency patterns, such as "Jarvis," to prevent unnecessary cloud API calls and preserve user privacy. Upon activation, raw audio is captured via the Speech Recognition library, undergoing acoustic modelling and noise-reduction filtering before being dispatched to the Gemini 2.5 Flash API. This "brain" of the module identifies user intent and entities, while the interaction loop is closed using the pyttsx3 text-to-speech engine to provide a hands-free, conversational experience.
2) Hand Gesture Recognition and System Control: Module 2 represents the motor skills and vision of the assistant, enabling a touchless interface that allows users to manipulate the operating system through spatial movements. The core of this module is the MediaPipe Hands framework, which employs a high-fidelity palm detection and subsequent landmark model to identify 21 distinct 3D coordinates on each hand. By calculating the Euclidean distance and angles between points—such as the distance between the tip of the thumb and index finger—the system accurately determines the state of each finger to map gestures like a fist, open palm, or V-sign to specific system-level functions. Once a gesture is validated through a debounce filter to avoid accidental triggers, the system invokes low-level Python libraries, including PyAutoGUI for simulating mouse movements and keyboard shortcuts, and Subprocess for launching system-level executables.
3) Emotion Detection: Module 3 introduces affective computing into the project, transforming the webcam into an emotional sensor that treats the user as a human with a psychological state rather than just a source of commands. The process begins with face frontalization using OpenCV to detect and align the face, ensuring the eyes and mouth are levelled even if the user's head is tilted. This aligned face is then passed through a Deep Convolutional Neural Network via the DeepFace library to analyze micro-expressions, such as the tightening of the lips or the furrowing of the brow. The output is a probability vector across seven universal emotions, where the dominant emotion is identified and exported as a metadata tag to be used for adaptive behaviour in subsequent modules.
4) Adaptive Response Generation: Module 4 acts as the decision engine, representing the most advanced layer of the system where the results of all previous modules are fused together to create a context-aware response. This module operates on priority-based logic to ensure the AI's behavior is synchronized with the user's current mood. For instance, if a negative state such as sadness or anger is detected, the assistant pivots its response from simple command execution to providing empathetic feedback or playing "Mood Playlists" via the pygame mixer. The assistant maintains a continuous feedback loop, monitoring whether these adaptive responses successfully shift the user's emotion back to a neutral or happy state. This integration evolves the assistant from a simple voice-command tool into a proactive digital companion that perceives, understands, acts, and adapts to the human user.

V. IMPLEMENTATION

The Emotion-Based AI Assistant is implemented using a multi-threaded Python architecture to manage the high-frequency data streams required for real-time interaction. The system leverages a decoupled design where computer vision tasks, such as hand tracking and facial analysis, are isolated into high-priority threads to ensure that the mouse cursor and system controls remain responsive without lag. For intelligence and reasoning, the platform integrates the Gemini 2.5 Flash API to handle contextual natural language understanding, while the DeepFace library provides the deep learning framework for affective computing. System-level automation is achieved through libraries like PyAutoGUI and Pywin32, allowing the software to interact directly with the Windows API. This implementation ensures a privacy-preserving and locally executable environment, as all visual and auditory processing is handled on the user's hardware rather than being stored on external servers.

VI. CONCLUSION

The Emotion-Based AI Assistant successfully demonstrates the application of affective computing and real-time computer vision to streamline human-computer interaction and enhance desktop productivity. The platform improves efficiency by providing a touchless interface for system control, minimizes interaction friction through contextual reasoning, and encourages an engaging user experience. By integrating empathetic responses into the core architecture, the assistant addresses the psychological needs of the user, creating a more responsive and human-centric digital environment.

REFERENCES

[1] Kapoor and S. R. Gupta, "Designing Emotion-Aware AI Assistants for Desktop Environments," International Conference on Artificial Intelligence and Human-Computer Interaction (ICAHCI), pp. 12-18, 2022.
[2] R. Patel and M. Singh, "Voice-Controlled Personal Assistants: A Survey on AI Integration and System Automation," International Journal of Intelligent Systems, vol. 35, no. 4, pp. 250-260, 2021.
[3] L. Chen and J. Zhao, "Real-Time Emotion Detection Using Facial Recognition in AI Applications," IEEE Transactions on Affective Computing, vol. 13, no. 2, pp. 450-462, 2022.
[4] H. Kim and T. Lee, "Dynamic OS Control through Natural Language Processing in Personal Assistants," International Journal of Software Engineering and Knowledge Engineering, vol. 31, no. 5, pp. 715-730, 2021.
[5] Smith et al., "Advances in Voice-Activated AI Assistants," Contextual Reasoning Survey, 2021.
[6] Brown et al., "Emotion Recognition in Human-Computer Interaction," Affective Computing Research, 2021.