Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

Gesturix: A Secure Multimodal Desktop Assistant Integrating Face Authentication, Voice Interaction, And Gesture Control

Sarthak Chaurasiya[1], Ayush Panwar[2], Pawan Yadav[3], Prof. Hridesh Gupta[4]

[1,2,3]Authors, [4]Mentor
Computer Science and Information Technology
Dronacharya Group of Institutions

Abstract

The field of Human-Computer Interaction (HCI) has now progressed to another level with developments in Artificial Intelligence, Computer Vision, and Natural Language Processing. The traditional systems require significant keyboard and mouse interaction. Therefore, they suffer from lack of accessibility and efficiency. This paper introduces GESTURIX, which is an intelligent desktop assistant and implements facial authentication, voice interaction, and gesture commands all in one system. The proposed system uses OpenCV library facial recognition for secure access management, speech recognition and text-to-speech functionality for two-way interaction, and MediaPipe library-powered hand gesture recognition for gesture commands and mouse and keyboard functionality. The system uses layered and event-driven design to ensure non-blocking and efficient execution.

Keywords: Desktop Assistant, Face Recognition, Voice Assistant, Gesture Control, Human–Computer Interaction, Computer Vision

1. Introduction

There has been a major evolution in Human Computer Interfaces because of the advancements taking place in artificial intelligence, computer vision, and speech processing technology. There has been a lot of dependence upon input devices such as keyboards and mouse with desktop computing, which might affect efficiency to some degree. With the advancements taking place in intelligent computing, the demand has risen for interacting with machines in a natural way with the help of Human Computer Interfaces.

Voice assistants are commonly used in the mobile and smart home systems context; nevertheless, the application of voice assistants in the desktop context is limited. The application of desktop assistants is limited to automated processes. There are no advanced security systems in the current desktop assistants. Moreover, the emerging issue of inappropriate system access looms large. The

conventional password security systems are insufficient.

In this article, the idea of GESTURIX has been introduced, a secured desktop assistant that incorporates facial recognition, voice communication, and gesture commands in a comprehensive system. The new method integrates various computer interfaces, thus improving usability, but security is also taken into consideration. In this article, the potential of computer vision and speech analysis has been successfully utilized in a real-time desktop system.

2. Related Work

There have been a few studies on voice interaction systems in an effort to improve computing system usability. Voice interaction systems using hidden Markov models and deep learning models are popular due to high accuracy levels. Even if there is an ability to use a computing system without using one's hands, it does not interface with another interaction technology. Also, there is no valid authentication mechanism in place.

Face recognition methods have also been extensively explored for biometric authentication. Eigenface, Fisherface, and Local Binary Pattern Histograms (LBPH) methods have been observed to give efficient results for face recognition and have been demonstrated to be efficient for processing in real time. The Haar Cascade classifiers are still in popular usage for methods of face recognition.

There has been a recent interest in gesture recognition systems because of the advent of lightweight platforms, such as MediaPipe. As indicated by previous research, the concept of having mouse and keyboard control using hand gestures has been proven; however, until now, they are mostly done as a separate application. GESTURIX contains the capabilities of voice, facial, and gesture recognition within a single platform, offering a more comprehensive and secure way of interaction.

3. System Overview

The design of the GESTURIX system lays emphasis on being modular and extendable to function as a desktop aid for secure access and multiple interaction tasks. The proposed system works on the idea of constantly monitoring user inputs in terms of voice commands, facial expressions, and hand gestures. Every interaction task is managed by a different module.

For its operations to begin, it first identifies the user using facial recognition. After which, if there is success in identification, then begins the process for voice command and gesture control. For seamless operations of all its features, this system supports a web-based graphical user interface for its operations. The system can work with efficient operations on consumer hardware.

4. Proposed Architecture

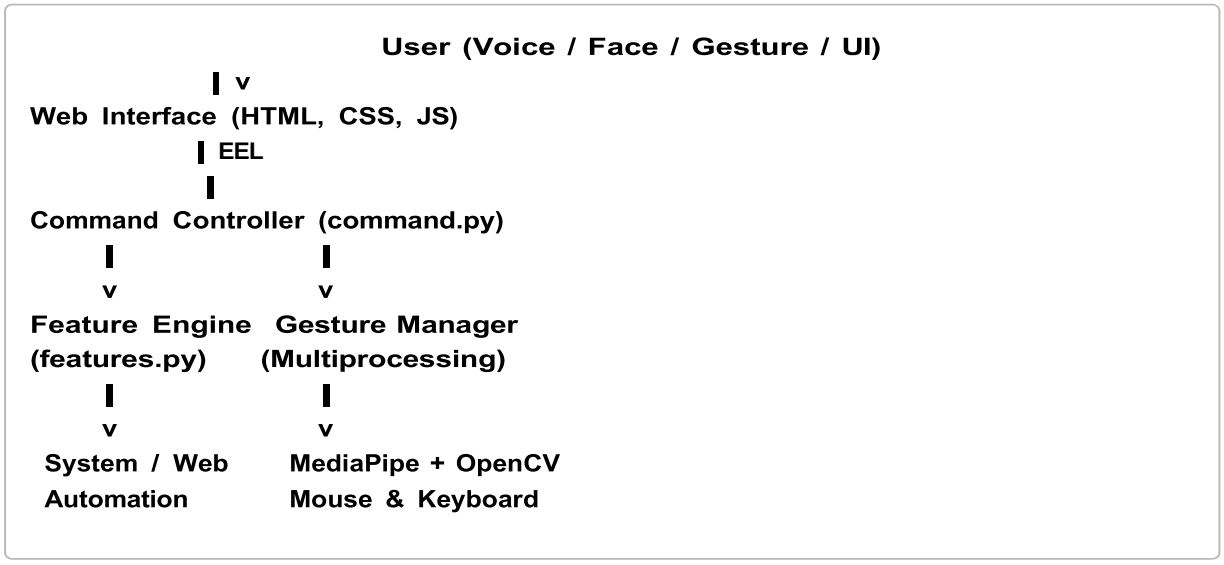
A. Architectural Design

The architecture of GESTURIX is composed of multiple layers:

1. Presentation Layer – Web-based UI developed using HTML, CSS, and JavaScript
2. Interface Layer – EEL framework enabling Python–Web communication
3. Command Processing Layer – Interprets and routes user commands
4. Feature Engine Layer – Executes system and web-related operations
5. Gesture Control Layer – Handles gesture recognition and automation
6. Face Authentication Layer – Verifies user identity using computer vision
7. Data Layer – Stores commands and contacts using SQLite
8. System Interaction Layer – Interfaces with OS-level resources

B. Block Diagram

Fig. 1. Overall System Architecture of GESTURIX. The figure illustrates the layered architecture of the proposed desktop assistant, showing the interaction between the user interface, command processing layer, feature engine, gesture control module, face authentication system, and underlying operating system resources.



5. Face Authentication Module

The face authentication module ensures secure access to the assistant. It uses: - Haar Cascade Classifier for face detection - LBPH algorithm for face recognitionThe authentication process involves capturing live video frames, detecting facial regions, and comparing them against a trained dataset. Access to the assistant is granted only upon successful recognition, enhancing system security.

6. Voice Interaction Module

Voice interaction is done with SpeechRecognition, which does the speech-to-text processing, and pyttsx3, which does the text-to-speech work. The system should be able to work with natural language commands to start applications, search on the internet, play media files, enable/disable gesture modes.

Hotword detection courtesy of using the Porcupine library enables the user to invoke the assistant without the use of hands because it supports continuous background listing with only a slight performance loss.

7. Gesture Control Module

Gesture control is enabled using 'MediaPipe Hands' for real-time hand landmark detection. The module provides the following features: Mouse control through single-hand operation Keyboard operation via gesture based on finger count

The separated process is used for gesture recognition so that it may be processed efficiently. For this, Python's “multiprocessing” module has been used. Either by voice commands or using buttons, turning the gesture mode on/off does not hamper other operations.

8. Database Management

The database is an SQLite database that contains the following data:

System application paths, Web URLs, Messaging contact information. Using this type of model makes the application highly extensible and configurable.

9. Results And Discussion

C. Performance Evaluation

The performance of the proposed GESTURIX system has been evaluated in terms of start-up time delay, authentication response time, command execution time delay, and overall performance. At first, there were problems of start-up time delay and calling of the facial authentication module. The start-up time delay problems were mitigated with asynchronous start-up implementation and model caching.

Experiments have revealed that the optimized system requires a time of 1-2 seconds for initializing the interface, while face authentication requires a time of 2-3 seconds in a normal lighting scenario conditions. Voice recognition and reaction rates are close to real time, taking an average of less than 500 milliseconds.

D. Resource Utilization

The CPU and memory usage are also being traced while the system is running. The structure of the system does not allow the computations involved in the gesture or facial authentication to hinder the normal

operation of the system. This is achievable using self-contained computations. C. User Experience Analysis For the user experience, the incorporation of the multimodal interface is a significant plus. It is easy to navigate between the operation using voice commands, gesture mechanism, and the UI. The facial authentication mechanism brings a significant security plus without hindering the user experience. It is observed in the experiment test that it is highly effective in a realistic setting. The accuracy rates are extremely high in the facial authentication tasks in ideal lighting conditions, the voice commands are processed without a lag, and the gesture mechanism enables the seamless operation of the mouse pointer and keyboard. This is achievable based on the module structure, which is explained above.

E. User Experience Analysis

In terms of usability, having multimodal interaction built right into the system makes it very user-friendly. The user has the option of switching between commands executed by speech, commands executed by gestures, and commands executed by UI elements. The face authentication feature that is secure and private protects users' data.

Experiments have demonstrated that the system works well and efficiently. The face recognition system works well when sufficient light is on, commands are processed with low latency using voice control, and gesture control is useful for controlling the cursor and keyboard. The system is well-organized and designed to prevent any malfunction from spreading to other parts of the system.

10. Limitations

- Face recognition performance depends on lighting conditions
- Speech recognition requires an active internet connection
- Gesture accuracy varies with camera quality and background noise
- WhatsApp automation is sensitive to UI changes

11. Future Scope

There exist several areas where the proposed system can be improved further in the future. The system can be implemented with the use of advanced models based upon deep learning technologies like Face recognition using FaceNet or ArcFace to increase the accuracy rate in difficult situations. Gesture control functionality can be improved to recognize dynamic gestures or enable user-defined gestural control. Multilingual speech recognition functionality can be added to improve accessibility.

The incorporation of cloud services will offer synchronization of profiles. The project may also be extended for controlling IoT devices. Thus, GESTURIX will be able to act as a central controller of smart environment.

Future developments may include:

- Face recognition using deep learning

- Learning gestures

- Support for multiple languages in voice
- Emotion recognition
- Connectivity to mobile and IoT
- Cloud profiles for users

12. Conclusion

This paper presented GESTURIX, a secure and intelligent desktop assistant that integrates face authentication, voice interaction, and gesture control into a unified system. By adopting a layered and modular architecture with multiprocessing support, the system achieves real-time performance, enhanced security, and improved usability. The proposed approach demonstrates the potential of multimodal interaction in next- generation desktop environments.

References

1. Viola, P., & Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features.
2. Ahonen, T., Hadid, A., & Pietikäinen, M. (2006). Face Recognition with Local Binary Patterns.
3. Zhang, F., et al. (2020). MediaPipe Hands: On-device Real-time Hand Tracking.
4. Jurafsky, D., & Martin, J. (2021). Speech and Language Processing.
5. OpenCV Documentation. (2024).
6. MediaPipe Framework Documentation. (2024)