
ISSN: 2321-2152
IJMECE
International Journal of modern electronics and communication engineering
E-Mail
editor.ijmece@gmail.com
editor@ijmece.com
www.ijmece.com

IJMECE
ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

AN INTEGRATED INTELLIGENT FRAMEWORK FOR VOICE-ASSISTED AND VIRTUAL MOUSE INTERACTION

Dr. K. Sathish[1]
Professor
Department of CSE(DS)
TKR College of Engineering and Technology
komatisathish459@gmail.com

Md.Faisal Pasha[2]
B.Tech(Scholar)
Department of CSE(DS)
TKR College of Engineering and Technology
mdfaisal3106@gmail.com

V.Rama Krishna Reddy[3]
B.Tech(Scholar)
Department of CSE(DS)
TKR College of Engineering and Technology
ramuvennapusa72@gmail.com

M.Durga Reddy[4]
B.Tech(Scholar)
Department of CSE(DS)
TKR College of Engineering and Technology
durgareddy9307m@gmail.com

P.Rahul[5]
B.Tech(Scholar)
Department of CSE(DS)
TKR College of Engineering and Technology
rahulpamu2003@gmail.com

ABSTRACT

The Voice Assistant And Gesture Controlled Virtual Mouse simplifies computer interaction by allowing users to control their devices with hand gestures and voice commands, reducing the need for physical contact. Using advanced Machine Learning and Computer Vision techniques, this system creates an easy way to perform tasks virtually without requiring extra hardware. It's designed to work smoothly on Windows and is adaptable to various user needs. The system is implemented to accurately detect both simple and complex hand gestures. It includes two main components: one that recognizes gestures directly using Hand Detection and another that works with gloves of any single color. This dual approach enhances its flexibility and usability. Gesture Recognition supports a wide range of functions, such as moving the cursor, clicking, scrolling, dragging, dropping, selecting multiple items, and controlling volume and brightness. These features help users interact with their computers more efficiently and reduce reliance on traditional input methods. The project also features a voice assistant named Echo. Echo allows users to manage the gesture recognition system, perform Google searches, find locations on Google Maps, navigate files, check the date and time, and handle tasks like copying and pasting. With voice commands, users can easily activate or deactivate Echo, making the system even more user-friendly. Together, the Gesture Controlled Virtual Mouse and Echo, offer a glimpse into the future of touch less computer control.

Keywords: Gesture Recognition, Voice Assistant, Machine Learning, Computer Vision, Hand Gesture Detection.

I. INTRODUCTION

In an era of rapid technological advancements, the way humans interact with computers is continuously evolving. The Voice Assistant and Gesture Controlled Virtual Mouse project is a step forward in redefining human-computer interaction by introducing a touch less, intuitive interface. This system allows users to control their devices using hand gestures and voice commands, thereby reducing the reliance on physical contact with traditional input devices like keyboards and mice. By employing advanced Machine Learning and Computer Vision techniques, the project offers a seamless and efficient method of performing essential tasks virtually, without the need for additional hardware. Designed to function smoothly on the Windows platform, this system is adaptable to diverse user needs, making it both practical and user-friendly.The project is implemented to detect both static and dynamic gestures accurately, enabling a wide range of functionalities. The system operates through two key modules: one that directly recognizes gestures using MediaPipe Hand Detection, and another that works with gloves of any uniform color. This dual approach ensures flexibility and robustness, accommodating various user scenarios. In addition to gesture recognition, the project features a voice assistant named Echo, which enhances the overall functionality. Proton allows users to manage the gesture recognition system, perform Google searches, locate places on Google Maps, navigate files, and execute various other tasks such as copying, pasting, and system control operations. Together, the Gesture Controlled Virtual Mouse and Echo present a sophisticated system that simplifies computer interaction and offers a glimpse into the future of touch less technology.

This system provides a practical solution to several challenges faced in conventional input methods. The gesture recognition component supports essential operations such as cursor movement, left and right clicks, scrolling, dragging, and dropping, along with advanced functionalities like volume and brightness control. These features enhance user efficiency and reduce dependency on traditional hardware. Meanwhile, the voice assistant complements the gesture recognition module by enabling users to control and navigate the system through voice commands, making the interface even more accessible. Proton is designed to launch or stop gesture recognition, retrieve the current date and time, navigate files, and execute searches, further improving the overall user experience.

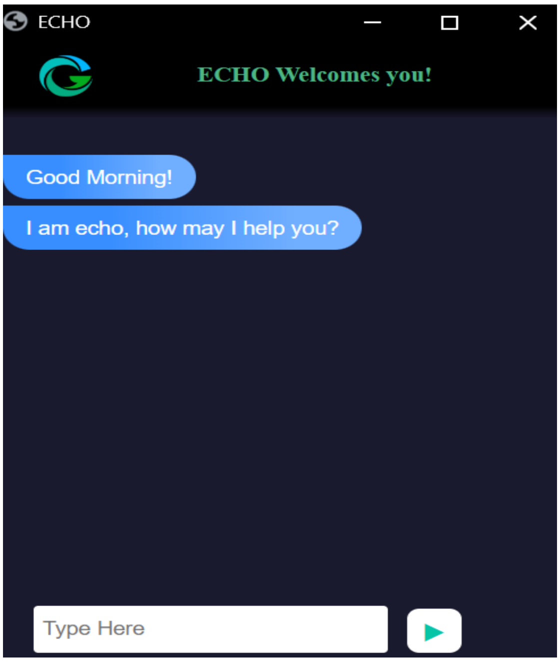


Fig. 1: Voice Assistant.

The technical backbone of this project includes state-of-the-art frameworks and tools. The system relies on MediaPipe's Hand Detection, powered by Convolutional Neural Networks (CNNs), implemented using pybind11 for precise and efficient gesture tracking. This allows for real-time recognition and interaction without the need for expensive external devices. Additionally, the integration of glove-based recognition ensures adaptability in diverse environments, offering consistent performance across different scenarios.

Overall, the Voice Assistant and Gesture Controlled Virtual Mouse is more than just an input system; it is a comprehensive solution that bridges the gap between humans and technology. With applications ranging from accessibility for individuals with physical disabilities to enhanced interaction in healthcare, gaming, education, and remote work environments, this project highlights the potential of touch less technology. By leveraging cutting-edge algorithms and user-centric design, it not only simplifies interaction but also showcases the possibilities for future innovations in human-computer interaction.

II. RELATED WORK

In the field of touch less interaction technologies, numerous advancements have been made that align with the objectives of the Voice Assistant and Gesture Controlled Virtual Mouse project. Researchers and developers have explored gesture recognition and voice command systems to enhance human-computer interaction, focusing on improving accessibility, efficiency, and user experience.Gesture recognition systems have been widely studied for their potential to replace traditional input devices. For instance, frameworks like MediaPipe, which this project employs, have been extensively used for real-time hand tracking and gesture detection. MediaPipe's robust hand landmark models have been integrated into various applications, including gaming interfaces, virtual reality systems, and accessibility tools for users with physical limitations. Similarly, Convolutional Neural Networks (CNNs) have been the cornerstone of many gesture recognition systems, enabling precise and scalable solutions across diverse use cases. Studies have shown that combining CNNs with real-time tracking frameworks enhances gesture recognition accuracy, making such systems adaptable to various environments.Voice assistants have also become an integral part of modern computing, with systems like Amazon Alexa, Google Assistant, and Apple Siri setting benchmarks for voice interaction. These systems leverage Natural Language Processing (NLP) to interpret commands, perform tasks, and provide a hands-free interaction mode. However, unlike standalone voice assistants, projects such as the Echo Voice Assistant integrate voice commands with gesture recognition, offering a multimodal interaction framework that enhances usability and user experience. Previous research in this area highlights the benefits of combining voice and gesture modalities to create intuitive and dynamic control systems.More over, advancements in wearable technologies, such as single-color gloves for gesture recognition, have further expanded the capabilities of touchless interfaces. Studies have demonstrated how wearable devices can improve gesture tracking accuracy in environments where hand landmarks are difficult to detect due to lighting conditions or background interference. This dual-module approach—direct hand detection and glove-based detection—has been explored in several research papers, showcasing its ability to ensure robustness and flexibility in diverse scenarios.Applications of such systems have been particularly impact in accessibility, healthcare, and gaming. For example, touch less systems have been employed in sterile environments like operating rooms to minimize physical contact with devices. In gaming, gesture-based controls offer immersive experiences, and in accessibility, they empower users with disabilities to interact with computers more effectively.

III. METHODOLOGY

The methodology for implementing the Voice Assistant and Gesture Controlled Virtual Mouse system is structured into multiple stages, each focusing on distinct aspects of development, integration, and testing.

A. System Design

This stage involves identifying the technical requirements and conceptualizing the system's design.Establish the goals of the project, such as enabling gesture-based control and voice command functionality.Ensure that the system operates without requiring additional external devices, making it accessible and cost-effective.Choose Windows OS as the primary platform for implementation due to its widespread usage.Select tools and frameworks such as Python 3.8.5, MediaPipe, CNNs, and pybind11 for the development.

B. Data Collection and Preprocessing

To implement robust gesture and voice recognition, data is collected and prepared:

Gesture Data: Use pre-trained models from MediaPipe Hand Detection to recognize and track hand landmarks.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

Voice Data: Utilize speech-to-text processing libraries and NLP models for voice recognition and command interpretation.

Preprocessing: Normalize the input data (e.g., scaling hand landmark coordinates) to ensure consistent performance across varying conditions.

C. Gesture Recognition Module Development

This module is responsible for interpreting static and dynamic hand gestures. Implement MediaPipe's hand landmark detection to identify gestures without any additional hardware.Develop a module for tracking gestures using gloves of any uniform color for scenarios where direct hand recognition may face challenges (e.g., poor lighting).Map gestures to corresponding actions such as cursor movement, left/right clicks, scrolling, drag-and-drop, and volume/brightness control.

D. Voice Assistant (Echo) Development

The voice assistant component, Proton, enhances the system's usability.Integrate speech-to-text models to interpret user commands.Implement functionalities such as launching/stopping gesture recognition, Google searches, file navigation, and checking the current date and time. Allow users to activate or deactivate the system, making it adaptable to different scenarios.

E. Integration of Modules

The gesture recognition and voice assistant modules are integrated to provide a cohesive user experience.Synchronize gestures and voice commands to avoid conflicts in simultaneous inputs. Design an intuitive interface to allow users to switch between gesture and voice-based interaction. Enable the system to execute complex tasks by combining gestures and voice commands (e.g., file navigation and selection).

F. Testing and Optimization

Rigorous testing ensures the reliability and efficiency of the system. Evaluate the accuracy of hand detection and gesture mapping under various conditions (e.g., lighting, backgrounds). Assess the reliability of speech recognition for different accents, tones, and environmental noises. Optimize the system for low-latency performance to ensure real-time interaction. Implement mechanisms to handle false gesture or voice command detections.

G. Deployment and Evaluation

Finally, the system is deployed and evaluated for real-world usability. Make the system available for Windows users, ensuring compatibility with a wide range of devices. Gather feedback from users to identify areas of improvement. Test the system in specific applications such as accessibility, gaming, and remote work to validate its versatility and effectiveness.

IV. IMPLEMENTATION DETAILS

In this section, we describe the implementation of the system, detailing the steps taken from data collection to model training and evaluation.

A. System Design

Define Objectives: Set clear goals for the system, such as enabling touchless control via gestures and voice.

Identify Requirements: Determine hardware and software requirements (e.g., camera, microphone, Python libraries).

Architecture Design: Create high-level architecture diagrams outlining major system components.

Feasibility Analysis: Analyze any limitations in terms of performance, hardware, and data input.

B. Data Collection and Preprocessing

Gesture Data Collection: Gather hand gesture data using MediaPipe’s hand tracking model.

Voice Data Collection: Record audio data via a microphone for speech recognition tasks.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

Data Preprocessing: Clean and normalize data (e.g., filter noise, resize images).

Data Labeling: Label gesture and voice data for training the models.

C. Gesture Recognition Module Development

Model Selection: Choose appropriate machine learning models (e.g., CNNs) for gesture recognition.

MediaPipe Integration: Use MediaPipe for real-time hand tracking and gesture recognition.

Gesture Classification: Train the model to detect specific gestures like cursor movement and clicking.

Optimization: Fine-tune the model to improve accuracy and reduce false positives.

D. Voice Assistant Development

Speech Recognition: Use tools like Google Cloud's Speech-to-Text for converting voice into commands.

Natural Language Processing: Implement NLP to process voice commands (e.g., searching, file navigation).

Voice Command Handling: Develop functions for executing commands like opening files and controlling system settings.

Text-to-Speech Integration: Implement a TTS engine for giving audio feedback (e.g., for system status).

E. Integration of Modules

Module Integration: Combine the gesture recognition and voice assistant into a single system.

Data Flow Management: Ensure smooth communication between the gesture and voice components.

Central Control Hub: Develop a central hub for managing inputs from both gestures and voice commands.

Synchronization: Ensure real-time coordination between modules for smooth interaction.

F. Testing and Optimization

Unit Testing: Test each individual module (gesture recognition, voice assistant, etc.) for functionality.

Integration Testing: Test the complete system to verify the interaction between modules.

Performance Evaluation: Assess the system's responsiveness, latency, and accuracy.

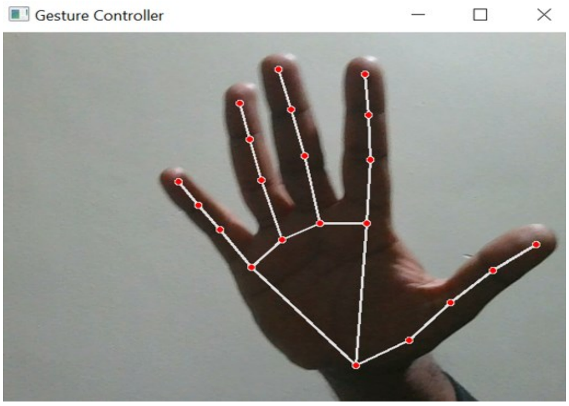
Bug Fixing and Optimization: Address issues identified during testing and optimize performance.

V. PROPOSED SYSTEM

The proposed system, Voice Assistant and Gesture Controlled Virtual Mouse, aims to revolutionize human-computer interaction by enabling touchless control using hand gestures and voice commands. By integrating Machine Learning and Computer Vision techniques, the system will allow users to perform various tasks without the need for traditional input devices like a mouse or keyboard. The system will utilize MediaPipe's hand detection model for accurate gesture recognition and speech recognition algorithms to interpret voice commands. This combination will offer users a seamless, intuitive experience for interacting with their devices.

The system will be designed to support both simple and complex gestures, such as cursor movement, clicking, scrolling, and file navigation. Additionally, it will feature a voice assistant, allowing users to perform tasks like Google searches, finding locations, checking the time, and managing the gesture recognition system. The system will be adaptable to various environments, ensuring optimal performance under different lighting and noise conditions.

This approach will not only enhance accessibility for individuals with physical impairments but also provide a more ergonomic and efficient way of interacting with computers, with potential applications in fields such as healthcare, education, and gaming.



VI. LITERATURE SURVEY

Paper 1: Accuracy Enhancement of Hand Gesture Recognition Using CNN: This research addresses challenges in hand gesture recognition by combining 2D-FFT and CNNs. Using Ultra Wide Bandwidth (UWB) radar to capture image data, the system transforms it with 2D-FFT and classifies it through CNNs. The proposed method shows faster learning time and similar accuracy compared to prominent models, improving human-machine interaction.

Paper 2: Gesture Recognition With Ultrasounds and Edge Computing:This work demonstrates the development of a gesture recognition system using ultrasonic signals and Edge devices. By employing Time of Flight (ToF) signals from two transceivers, the system recognizes seven gestures with accuracy between 84.18% and 98.4%. The approach processes data locally on the device, eliminating the need for external services.

Paper 3: Glove-Based Hand Gesture Recognition for Diver Communication: A smart dive glove recognizing 13 hand gestures for diving communication was developed using five dielectric elastomer sensors and machine learning classifiers. The classifiers—Decision Tree, SVM, Logistic Regression, Gaussian Naïve Bayes, and Multilayer Perceptron—showed accuracy between 0.95 and 0.98 in dry conditions, with a slight performance drop underwater due to environmental factors.

VII. CONCLUSION AND FUTURE WORK

The Voice Assistant and Gesture Controlled Virtual Mouse system successfully demonstrates the potential for touchless interaction with computers using a combination of gesture recognition and voice commands. By leveraging Machine Learning and Computer Vision, the system enables users to perform common tasks such as cursor movement, clicking, and scrolling, all through intuitive gestures and voice instructions. This innovative approach not only provides a more ergonomic and efficient method of interaction but also enhances accessibility, particularly for individuals with mobility impairments.While the system performs well under controlled conditions, certain challenges remain, such as reduced performance under non-optimal lighting or noisy environments. Nevertheless, the dual-input approach offers flexibility, allowing users to switch between gestures and voice commands as needed. The system’s ability to adapt to various user needs and its potential applications across industries such as healthcare, education, and entertainment makes it a promising tool for the future of human-computer interaction.In the future, the system can be improved by integrating advanced gesture recognition techniques, including 3D hand tracking and the use of depth sensors to enhance accuracy. The speech recognition module could be optimized for noisy environments, using noise cancellation algorithms. Additionally, expanding the range of recognized gestures, including complex multi-finger movements and more voice commands, would further increase the system’s capabilities. Integrating the system with smart home devices and virtual reality environments could open new possibilities for touchless control. Furthermore, optimizing the system to work on low-end devices will broaden its accessibility to a wider user base.

VIII.ACKNOWLEDMENTS

I would like to extend my heartfelt gratitude to the Dr.K.Sathish(Professor), my Guide at TKR College of Engineering Technology, for his unwavering guidance, expertise, and continuous support throughout the course of this research. His insightful feedback, constructive criticism, and dedication were pivotal to the successful completion of this project.

IX. REFERENCES

[1] E. T. Lau, L. Sun, and Q. Yang, ''Modelling, prediction and classification of student academic performance using artificial neural networks,'' Social Netw. Appl. Sci., vol. 1, no. 9, pp. 1–9, Sep. 2019.

[2] W. Wang, Y. Zhao, Y. J. Wu, and M. Goh, ''Factors of dropout from MOOCs: A bibliometric review,'' Library Hi Tech, vol. 41, no. 2, pp. 432–453, Jun. 2023.

[3] X. Li and S. He, ''Research and analysis of student portrait based on campus big data,'' in Proc. IEEE 6th Int. Conf. Big Data Analytics (ICBDA), Mar. 2021, pp. 23–27.

[4] F. Yang and F. W. B. Li, ''Study on student performance estimation, student progress analysis, and student potential prediction based on data mining,'' Comput. Educ., vol. 123, pp. 97–108, Aug. 2018.

[5] Z. Mingyu, W. Sutong, W. Yanzhang, and W. Dujuan, ''An interpretable prediction method for university student academic crisis warning,'' Complex Intell. Syst., vol. 8, no. 1, pp. 323–336, Feb. 2022.

[6] L. E. Adamova and O. O. Varlamov, ''Logic artificial intelligence application for the students individual trajectories introduction,'' in Proc. ITM Web Conf., vol. 35, 2020, p. 2001.

[7] A. Muhammad, Q. Zhou, G. Beydoun, D. Xu, and J. Shen, ''Learning path adaptation in online learning systems,'' in Proc. IEEE 20th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD), May 2016, pp. 421–426.

[8] H. Lim, S. Kim, K.-M. Chung, K. Lee, T. Kim, and J. Heo, ''Is college students' trajectory associated with academic performance?'' Comput. Educ., vol. 178, Mar. 2022, Art. no. 104397.

[9] L. Meng, W. Zhang, Y. Chu, and M. Zhang, ''LD–LP generation of personalized learning path based on learning diagnosis,'' IEEE Trans. Learn. Technol., vol. 14, no. 1, pp. 122–128, Feb. 2021.

[10] A. A. Mubarak, H. Cao, and W. Zhang, ''Prediction of students' early dropout based on their interaction logs in online learning environment,'' Interact. Learn. Environ., vol. 30, no. 8, pp. 1414–1433, Jul. 2022.

[11] N. Tomasevic, N. Gvozdenovic, and S. Vranes, ''An overview and comparison of supervised data mining techniques for student exam performance prediction,'' Comput. Educ., vol. 143, Jan. 2020, Art. no. 103676.

[12] B. K. Bhardwaj and S. Pal, ''Data mining: A prediction for performance improvement using classification,'' 2012, arXiv:1201.3418.

[13] Y. Zhao, Q. Xu, M. Chen, and G. M. Weiss, ''Predicting student performance in a master of data science program using admissions data,'' in Proc. Int. Educ. Data Mining Soc., Jul. 2020, pp. 1–9.

[14] G. Su-Hui, B. Cheng-Jie, and W. Quan, ''Hadoop-based college student behavior warning decision system,'' in Proc. IEEE 3rd Int. Conf. Big Data Anal. (ICBDA), Mar. 2018, pp. 217–221.

[15] A. Akram, C. Fu, Y. Li, M. Y. Javed, R. Lin, Y. Jiang, and Y. Tang, ''Predicting students' academic procrastination in blended learning course using homework submission data,'' IEEE Access, vol. 7, pp. 102487–102498, 2019.

[16] N. Hidayat, R. Wardoyo, A. Sn, and H. Dwi, ''Enhanced performance of the automatic learning style detection model using a combination of modified K-means algorithm and naive Bayesian,'' Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 3, pp. 638–648, 2020.

[17] N. T. Nghe, P. Janecek, and P. Haddawy, ''A comparative analysis of techniques for predicting academic performance,'' in Proc. 37th Annu. Frontiers Educ. Conf. Global Eng., Knowl. Without Borders, Opportunities Without Passports, Oct. 2007, p. 7.

[18] D. Kabakchieva, ''Predicting student performance by using data mining methods for classification,'' Cybern. Inf. Technol., vol. 13, no. 1, pp. 61–72, Mar. 2013.

[19] J. L. Rastrollo-Guerrero, J. A. Gómez-Pulido, and A. Durán-Domínguez, ''Analyzing and

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025
predicting students' performance by means of machine learning: A review,'' Appl. Sci., vol. 10, no. 3, p. 1042, Feb. 2020.

[20] R. Asif, A. Merceron, S. A. Ali, and N. G. Haider, ''Analyzing undergraduate students' performance using educational data mining,'' Comput. Educ., vol. 113, pp. 177–194, Oct. 2017.

[21] M. Yağcı, ''Educational data mining: Prediction of students' academic performance using machine learning algorithms,'' Smart Learn. Environ., vol. 9, no. 1, p. 11, 2022.

[22] J. Niyogisubizo, L. Liao, E. Nziyumva, E. Murwanashyaka, and P. C. Nshimyumukiza, ''Predicting student's dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization,'' Comput. Educ., Artif. Intell., vol. 3, Jan. 2022, Art. no. 100066.

[23] L. Huang, C.-D. Wang, H.-Y. Chao, J.-H. Lai, and P. S. Yu, ''A score prediction approach for optional course recommendation via cross-userdomain collaborative filtering,'' IEEE Access, vol. 7, pp. 19550–19563, 2019.

[24] A. Alshanqiti and A. Namoun, ''Predicting student performance and its influential factors using hybrid regression and multi-label classification,'' IEEE Access, vol. 8, pp. 203827–203844, 2020.

[25] M. Akour, H. A. Sghaier, and O. Al Qasem, ''The effectiveness of using deep learning algorithms in predicting students achievements,'' Indonesian J. Electr. Eng. Comput. Sci., vol. 19, no. 1, p. 388, Jul. 2020.

[26] V. L. Miguéis, A. Freitas, P. J. V. Garcia, and A. Silva, ''Early segmentation of students according to their academic performance: A predictive modelling approach,'' Decis. Support Syst., vol. 115, pp. 36–51, Nov. 2018.

[27] W. Chango, R. Cerezo, M. Sanchez-Santillan, R. Azevedo, and C. Romero, ''Improving prediction of students' performance in intelligent tutoring systems using attribute selection and ensembles of different multimodal data sources,'' J. Comput. Higher Educ., vol. 33, no. 3, pp. 614–634, Dec. 2021.