Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/22977
DOI URL: http://dx.doi.org/10.21474/IJAR01/22977

RESEARCH ARTICLE

AUREVIX: A SMART GESTURE-CONTROLLED CENTER WITH MULTI-MODE RECOGNITION AND VOICE INPUT FOR HUMAN-COMPUTER INTERACTION

Pratha More[1], Shirisha Vakiti[1], Priti Ghodke[2], Komal Nehete[3] and Hemant Bansal[2]

1. Student, Department of Computer Engineering, St. John College of Engineering and Management, Palghar.
2. Lecturer, Department of Computer Engineering, St. John College of Engineering and Management, Palghar.
3. Lecturer, Department of Information Technology, St. John College of Engineering and Management, Palghar.

Manuscript Info

Manuscript History
Received: 10 January 2026
Final Accepted: 12 February 2026
Published: March 2026

Key words:-
Gesture Recognition, Voice Recognition, Human-Computer Interaction (HCI), Multi-Modal Interaction, Machine Learning, Computer Vision, Sensor-Based Control, Artificial Intelligence (AI), Speech Processing, Image Processing, Real-Time System, Automation, Hand Gesture Detection, Natural User Interface (NUI), Deep Learning

Abstract

Nowadays the demand of touchless human-computer interaction system is increasing, where traditional devices like mouse and keyboard are not convenient in every situation. In this paper a Smart Gesture-Controlled System with voice input named “Aurevix” is presented, which controls the computer with hand gestures and voice input. This system uses webcam and microphone to get inputs from users and through MediaPipe, OpenCV and speech processing technique detects gestures and voice commands. Aurevix supports multiple functionalitie s, like virtual mouse control, drawing, gaming, presentation control and gesture training.This system works in real-time with minimal delay and provides smooth interaction. Overall, this is cost-effective and user friendly solution, which useful for modern touchless computing.

Introduction:-

For many years, the relation between human and a computer has been connected through some of the physical devices. From the early days of mechanical trackballs to today’s advanced optical mouse, the way we interact with computer has not changed yet; it has only improved slightly day by day. User still requires sitting at a desk, using a flat surface and relying on physical hardware to control our machines. However, nowadays as technology and digital environment has become more advanced and three-dimensional, these available traditional tools are starting to feel outdated. Although there is lack of freedom, there are also some physical problems. The grip which is needed to hold the mouse is lacking in some of the people. In addition, after the global pandemic, there is the need of clean and touchless interfaces in places like, hospitals, laboratories, etc.Using computer vision technology, Aurevix can detect and track 21 important points of our hand, also called as landmarks, on the human hand in real time. These landmarks represent the joints and tips of the fingers. Instead of simply moving a cursor, the system understands the movement and shape of the hand. It calculates the distance between different fingers and parts of the palm using mathematical formula such as Euclidean distance. With this information, the app converts visual hand movements into computer commands. This makes the interaction feel natural and smooth, almost like communicating with the computer directly.What makes Aurevix different from basic gesture tracking systems as it is inbuilt with Multi-Mode Architecture? In the developmental phase of this app we came to know that different tasks require different levels of control and sensitivity. A designer drawing on screen needs very precise movements, while a gamer needs

Corresponding Author:-Priti Ghodke
Address:-Lecturer, Department of Computer Engineering, St. John College of Engineering and Management, Palghar.

fast and responsive action. Because of this, we have designed four different operating modes that are been adjust according to the user need thus making the system user efficient.

Desktop mode is designed for everyday tasks such as browsing the internet, opening files and general computer use. It focuses on smooth cursor movement and prevents accidental clicks while working.Gaming mode focuses on speed and performance. In this mode, the system reduces processing delays so that fast hand movements can quickly translate into gaming actions. Draw Mode is created for creative work such as digital art. It provides extremely stable and accurate tracking so user draws or paint in the air with high precision.The multi control mode is designed for long distance interaction. This allows the user to control voice assistant, presentations, volume, etc.One major challenge when controlling a mouse using a webcam is a problem known as cursor jitter. Because webcam can produce unwanted noise and human hands naturally make very small movements, the cursor may shake or more slightly even when the user tries to keep their hand still. To solve such issues we have developed a recursive smoothening algorithm. This algorithm works as a stabilizer. It filters out unwanted noise while keeping actual movement of the hand. Due to which the cursor moves smoothly and steadily. So Aurevix is an app which focuses on eliminating the physical devices and humans become the primary way to interact with computers.

Fig.1: Splash Screen

Significance of the Study:-

This study demonstrates a highly accurate gesture-controlled virtual mouse system using computer vision techniques, though it faces challenges in precise right-click operations and fine text selection [1].The proposed system integrates hand gestures with voice control to assist physically challenged users, but its performance is affected by lighting conditions and system latency [2].This research introduces a dual-mode gesture and voice-controlled system with advanced automation features, yet its application is limited to specific operating systems [3].The system effectively replaces traditional input devices using webcam-based tracking, although it encounters limitations in clicking accuracy and dragging precision [4].The study achieves good accuracy in gesture recognition using CNN and MediaPipe, but highlights the issue of high computational cost affecting real-time performance [5].This work presents a highly accurate gesture-controlled mouse with voice assistance, though further testing is required under diverse environmental conditions [6].The research focuses on enhancing system-level controls using gesture recognition, but suggests improvements in accuracy and feature expansion for better usability [7].The system adapts well to varying lighting and finger positions, yet requires further development for scalability and web-based implementation [8].This study improves gesture stability using filtering techniques, but still faces challenges in performing complex actions like dragging and right-clicking [9].The research emphasizes inclusive and responsive interaction systems, while recommending future enhancements such as depth sensing and adaptive learning[10].

Materials and Methods:-

Proposed System-Aurevix is architecture of high speed and Vision based controller that is designed to reduce the dependence of physical Human-Computer Interaction (HCI) devices such as mouse and keyboard. We build this

system to convert the raw hand gestures into mouse or keyboard action depending on the gesture, by shifting the complexity of spatial tracking to a dedicated software engine. Aurevix only require a standard 720p RGB camera.

Functional Architecture and Data PipelineThe internal logic of Aurevix is as followed:
I)Optical Acquisition and Adaptive Pre-processing: Firstly, OpenCV library is used to capture video input from the webcam. Captured frames are horizontally mirrored to make user interaction feel natural and intuitive, means the movement done by the user that should reflect on the screen. During development, one major challenge faced was "Landmark Drop-out", where system could not detect the gestures accurately in low-light conditions. To solve this issue, a Luminance Normalisation module was implemented. This module monitors the average pixel intensity of each frame. When lightning falls below a predefined threshold, the system automatically adjusts brightness and contrast. Through this adaptive pre-processing, gesture detection remians stable and reliable even in low-light scenarios.

II)Neural Interface and 21-Point Mapping: MediaPipe framework is used for gesture detection, which detects 21 unique landmark points for hand tracking. These landmarks create a digital hand model that enables accurate tracking of hand movements. The system is programmed in such a way that the index finger controls cursor movement. For basic interactions such as clicking, open palm gesture is assigned. A Single Shot Detector (SSD) is used for real-time hand detection, that identifies hand efficiently in a frame.Using this 21 landmark points, the system creates a spatial mapping structure that calculates the distance and angles between finger positions. Based on this analysis different gestures are defined and mapped to specific actions like click, scroll and navigation. This approach enables the system to achieve high precision and real-time responsiveness, which are critical for gesture-based interaction.

III) Coordinate Scaling and User Comfort: Another challenge we faced was the difference in resolution between the camera and the monitor. As the webcam capture video in 640 x 480, but most of the monitor screen are of 1920 x 1080. To solve Thiswe created an algorithm to convert the webcam feedback into monitor screen in proportional scaling, depending on the size of the monitor screen. This let cursor move around the screen more smoothly across the entire screen, regardless of the monitor's size.

IV) Temporal Smoothing and Jitter Mitigation: One of the biggest challenges was that the camera was even taking the smallest "jitter" or shaking of hand as mouse movement. To be accurate the project is having jitter issue wherein the cursor shakes a lot even if user hand is stable. This problem is occurring due to camera sensor noise and human hands are sometimes not perfectly stable, to solve this issue we used smoothing algorithm i.e., only current frame is not visible, rather the average of last 5 Frames are calculated, then according to weighted average the final cursor position is decided. As a Results cursor will move smoothly, no shaking occurs and give the feel of real physical mouse.

The final cursor position is calculated using the following equation:
$P_{final} = (\alpha. P_{current}) + ((1- \alpha).P_{average\_past})$
Where,
P_{final} = Final cursor position
$P_{current}$ = Current frame position
$P_{average\_past}$ = Average position of previous frame
α = Factor of smoothing used to balance the current and previous position

Context-Aware Multi-Mode Logic:
I) Standard Desktop Mode: Standard Desktop Mode is a default operation mode, where user wants to perform normal computer tasks. At this time system automatically works in this mode. The main purpose of this mode is to navigate files, web browsing, application opening and general mouse interaction. During the time of development one important problem is observed, i.e., when user moves their hand on screen sometimes the system performs unintentional clicks. This issue occurs when user moves their hand too fast or changes the gesture. Due to which camera sometimes records small finger movements as click gestures and accidental click occurs. To solve this problem. we implemented click-lock mechanism. This mechanism allows shaky hand movements so that tiny tap gestures can be ignored.

II) High-Performance Gaming Mode: In gaming environment, response time and system latency is a very critical factor. Traditional mouse devices provide direct hardware input, due to which its response time is extremely fast. But in gesture-based system, cursor movement is dependent on camera frame and image processing which causes processing delay. The main objective of this mode is to create responsive system, so that gaming interaction feels natural. In normal operating mode cursor stability is improved using smoothing algorithm and frame averaging. But in gaming environment more smoothing can sometimes leads to reaction delay. The smoothing buffer of this system is reduced or tightened, so that cursor movement reacts faster. After this optimization the average interaction latency of system is maintained by approximately 45-55 milliseconds applicable for casual gaming interaction. In this mode an interactive gesture mapping is added.

III) Intelligent draw mode: The Draw mode is designed for interactive Gesture-baseddrawing experience, in which user can perform gestures and create a digital artwork without using any physical device. This mode converts the system to virtual canvas, where user can draw in real time just by natural hands movements. In this mode the primary finger is the index finger, which is used in the form of drawing tool through which user can freely draw any kinds of design on screen. Different gestures are mapped with different drawing actions, so that interaction feels smooth and natural. For, example, five finger gesture movement acts as a duster to clear the canvas, rock gesture represents pen-up action, through which user can stop drawing temporarily. Other than this, gesture-based tool switching is also available, so that user can change tools without clicking on the menu. One major challenge we faced is to maintain accuracy and smoothness, because of unstable hand movements, the jitters were distorting the drawing. To solve this problem. we have used smoothing algorithm which stabilizes the input to give smooth output. This mode provides multiple tools and dynamic colour options, through with user can draw easily. System works in real-time with minimal delay, where interaction is smooth and feel is natural. This mode enhances the user's creativity and makes the system interactive-creative platform.

IV) Presentation Mode: Presentation Mode is specially designed so that user can control presentation efficiently without using physical device. In this mode user can perform simple hand gestures to change slides next to previous, zoom in/out media or to navigate to specific section. This hands-free interaction provides seamless experience, which is useful for classrooms, seminars and professional presentation. One of the important enhancements in this mode is that, draw mode is also integrated. With the help of this feature user can highlight and draw real-time diagram during the presentation. This functionality is useful, especially for teaching and explanation, so that presenter can visually represent their point and audience engagement can be improved. During the time of development the major challenge faced was accidental gesture detection, as user's normal hand movements were also mistakenly detected as gestures during presentations. To handle this issue gesture filtering, threshold control and delay-based validation techniques are been applied, through which only intentional gestures are executed and unwanted actions are avoided. With respect to performance, the system maintains minimal latency and provides real-time gesture recognition. Due to efficient processing, slide transition and drawing actions both works smoothly and responsively. Overall, this mode becomes more interactive, flexible and professional where the combination of user control and creativity meets.

V) Guided Gesture Training Mode: Training Mode is specially designed to make the user understand the systems predefined gestures and to learn how to perform them effectively. Because, gestures are developed by the developer at the backend, that are not directly known by the end user, that's why this mode provides platform, where user can practice and master all gestures. In this mode a training hub is integrated, in which multiple interactive modules are available, they are, Gesture Analytics, Reflex Challenge and Adaptive Trainer. Gesture Analytics module analyses the user's gesture accuracy, consistency and stability, on the basis of this provides feedback. Reflex challenge mode improves the user's reaction time and gesture recognition speed, so that real-time interaction becomes fast. Most important component of this mode is Adaptive Trainer, which works on AI-approach. This module continuously collects and analyses user's gesture data, through which user behaviour, moment pattern and performance trends can be understood. On the basis of this analysis, system can dynamically adjust its detection sensitivity, gesture thresholds and response logic. This adaptive mechanism ensures that, system gets personalized according to user's preference because every individual has different gesture styles. As user continues to practice different gestures, the system performance improves, due to which gesture recognition becomes more accurate, stable and responsive. The main objective of this mode is not only gesture practice but also to build coordination between user and system. Through continuous feedback, performance tracking and AI-based adaption Aurevix becomes smart, self-learning and user-centric platform. Overall, this mode converts the system into intelligent, adaptive and evolving interaction system from static gesture controller, which is highly efficient and reliable for long term usage.

VI) Voice Input: Aurevix system is integrated with intelligent voice input feature, which allows user, the flexibility to interact with gestures as well as voice commands. This feature acts as a virtual assistant, which understands user's speech input and performs actions accordingly. As soon as the microphone is activated, system automatically generates one context-aware greeting, which is on the basis of current time, like morning, afternoon or evening greeting. This functionality makes the system more interactive and human-like, which improves the user experience. At the backend of the system some commonly used voice commands are pre-defined, which user generally gives to virtual assistant. Voice Command Processor module processes user's speech and converts them to meaningful commands. Audio manager module efficiently handles the microphone input and captures smooth audio, whereas Active AI Engine manages command execution and response generation. This feature supports real-time voice processing, by which commands are quickly recognized and executed. The integration with gesture-based system creates a hybrid interaction environment, where users can use gestures and voice according to their convenience. This voice input functionality makes Aurevix system advanced, flexible and intelligent interaction platform, which provides both natural communication and efficient control.

System Architecture:
The system architecture Aurevix is designed in such a way that interaction between user and computer becomes seamless, real-time and touchless, in which both hand gestures and voice input can be used. This architecture follows a modular pipeline approach, where every stage has its specific role. By which whole system becomes efficient, scalable and responsive.

1) Input Stage: The first stage of system is to collect data inputs, where user gestures and voice commands are captured. For gesture input webcam is used, which continuously captures video frames and detects user hand movements. As well as, microphone receives voice input, by user can speak up their commands. These both inputs are fed into the system in a parallel manner, by which hybrid interaction is possible. The main objective of input stage is to collect raw data without any delay, so that real time can be maintained.

2) Processing Stage: In this stage captured input is converted to usable format. MediaPipe and OpenCV are used for gesture processing. Through hand-tracking MediePipe detects 21-point landmarks, by which the exact position and orientation of the fingers is identified. OpenCV processes the image frame and makes the detection smoother and efficient. For voice input speech recognition techniques are applied, in which audio signal are converted to text or command format. This stage transforms raw input to structured data, which is getting ready for further modules.

3) Gesture and Voice Processing Module: This is the core processing module of the system where actual intelligence works. At this stage a detailed analysis of hand landmarks takes place, in which finger positions, distances and angles are calculated. On the basis of these parameters different gestures are identified, like click, scroll, drag, drop, etc. Voice input is also interpreted, where system understands the intent of user's speech. On the basis of predefined command patterns system decides, which action to perform. The main objective of this module is, to process raw data and to convert it to meaningful instructions, by which accurate gestures and voice recognition can be achieved.

4) Command Mapping: In command mapping, detected gestures and voice input are mapped with actual system action. Every gesture and voice command has its own predefined function, like cursor movement, left click, right click or multimedia control. This stage ensures that, a correct relationship can be maintained between user input and system output. If mapping is not accurate, then system will perform false actions, due to which this stage is very critical.

5) Command Execution: In this stage mapped commands are executed. This system interacts with the operating system to perform real-time actions. In this cursor movement, clicking operations, scrolling, drag/drop and multimedia controls, like volume adjustment or play/pause are included. In execution stage the main focus is on speed and responsiveness. System has to execute commands at minimal latency, so that user experience feels smooth and natural.

6) Output Stage: In this final stage system provides output to the user, where user can control without using any physical device. This touchless interaction makes the system modern and efficient. In this stage, user actions are immediately reflected, like cursor movement or application response, by which system's real-time behaviour is maintained.

7) Integrated Hybrid Interaction: One of the most important features of Aurevix architecture is that, gesture and voice both inputs integrated. User can user gesture or voice according to their convenience, or can combine both. This hybrid interaction makes the system flexible and user-friendly, which suitable for different scenarios like, gaming, presentation and daily computing.

8) **System Advantages:**This architecture makes modular system design system highly scalable and maintainable. Every module works independently, by which future upgrades and feature additions are easily possible. Real-time processing, minimal latency and multi-mode support make Aurevix powerful and efficient gesture-based interaction system.

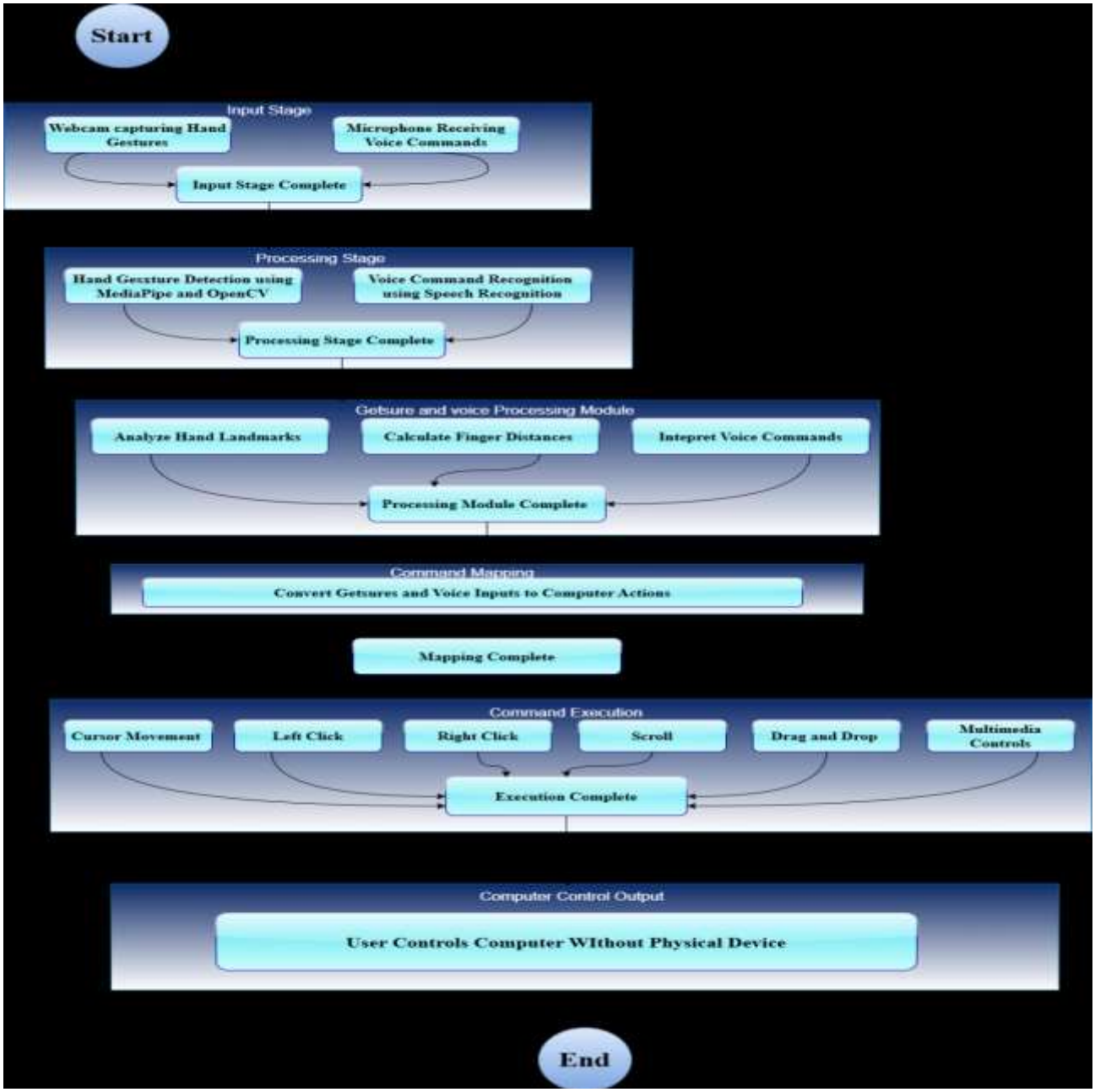

Fig.2: System Architecture

Methodology:-

The methodology of Aurevix is based on designing real-time, touchless human computer interaction system that uses both gesture recognition and voice input. System is divided into multiple stages so that the process from input capture to final output execution is efficient, accurate and has low latency.

Data Acquisition: In this stage system collects input from the user. Webcam continuously captures video frames, in which user's hand gestures are detected. These frames are passed from real-time process.
At the same time, microphone captures voice input, which allows user to give commands. Both gesture and voice inputs are processed in parallel, enabling hybrid interaction.

Image Pre-processing: Captured video frames are not directly captured, but they are pre-processed. By using OpenCV frames are resized, flipped and optimized, so that detection becomes fast and accurate.For maintaining noise reduction and frame consistency basic image processing techniques are applied, by which system performance is improved.

Hand Landmark Detection: For gesture detection MediaPipe framework is used, which detects 21 landmarks of user's hand. These landmarks represent finger joints and positions. These points are tracked in every frame, by which system gets real-time data of hand movements. This stage is the base of gesture recognition.

Gesture Recognition and Classification: By using detected landmarks gestures are identified. Finger positions, distances and angles are calculated, so that specific gestures can be detected.
• Cursor movements
• Click (left/right)
• Scroll
• Drag and Drop
• Drawing gestures
• Gaming action
For classification of gesture a rule-based logic is used, in which threshold and conditions are defined. To avoid unintentional gestures filtering technique and delay validation are applied.

Voice Command Processing: For voice input speech processing techniques are used. Audio manager handles the microphone input and captures continuous audio stream.Voice command processor processes the user's speech and converts it to test format. On the basis of predefined commands Active AI Engine identifies the user's intent decides action accordingly.

Command Mapping: In this stage recognized gestures and voice input are mapped with system-level actions. Every gesture or command has its own specific function, like cursor movement, mouse clicks or application control. This system is core linking mechanism which converts input to output

Command Execution: To execute mapped commands this system triggers the functions of operating system. This stage works in real-time where, cursor movement, clicking, scrolling, drawing and multimedia controls are performed.Execution speed and responsiveness is critical in this system, so that user can feel smooth and natural interaction.

Smoothing and Performance Optimization: In gesture-based system cursor instability and jitter are one of the common issues.

To solve this smoothing algorithmare implemented, which stabilizes the input.Differentmode has different optimizations:
• In normal mode requires more smoothing.
• In gaming mode to reduce latency, smoothing is reduced.
This provides balanced performance.

Learning Mechanism: Through training mode system analyses user's gesture pattern. Collect user's performance data and adjust the thresholds, sensitivity and detection parameters.This adaptive approach makes the system user-specific and further improves the accuracy.

Integration Of Multi-Mode System: Aurevix is a multi-mode system in which different functionalities like mouse control, drawing, gaming and presentation are integrated. Mode manger ensures that correct mode is activated and accordingly gesture mapping is applied.This makes the integration system flexible and versatile.

App Functionalities- Aurevix is a gesture-based human-computer interaction system that allows users to control the system without traditional input devices like mouse and keyboard. It includes multiple functional modes implemented in the application that target different real-world use cases.

Core functionalities include:

1. **Desktop Control:** Users can perform basic system operations like cursor movements, clicking, scrolling through hand gestures, which eliminate traditional mouse usage.

Fig3. Dashboard

2. **Gaming Interaction:** In gaming mode, Aurevix maps gestures with game controls, which gives user the advantage to play games interactively without physical controller.

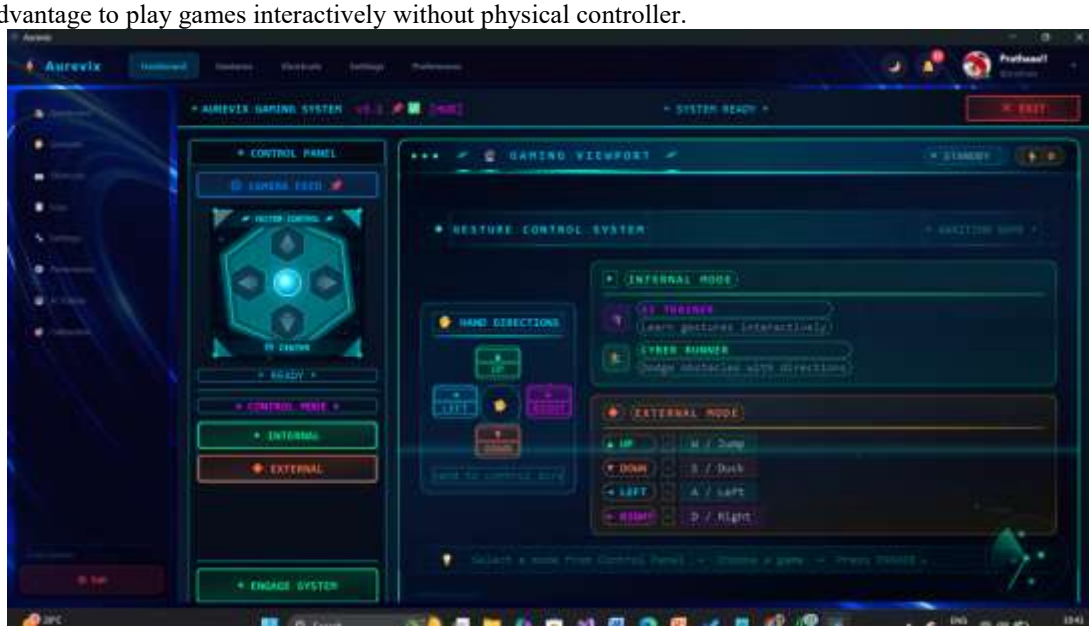

Fig4. Gaming System

3. **Draw Interface:** System uses real-time hand tracking to enable drawing and writing on digital canvas, which is useful for creative tasks.

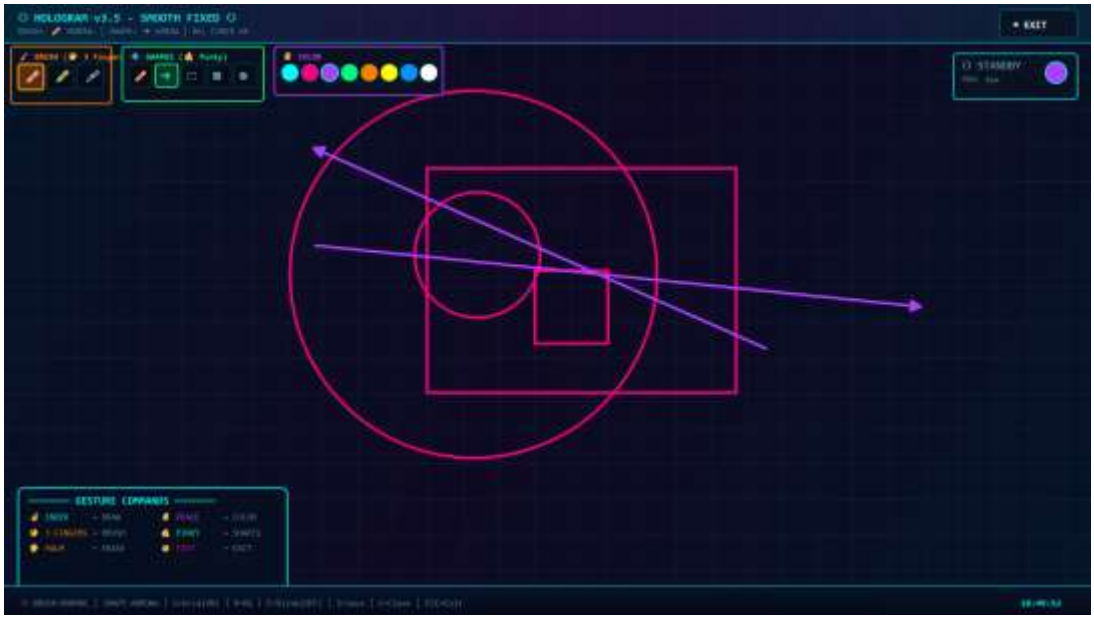

Fig5. Draw Mode

4. **Presentation Control:** Gesture-based slide navigation allows next/previous, which provides seamless control during presentation as there is no need of touching any device.

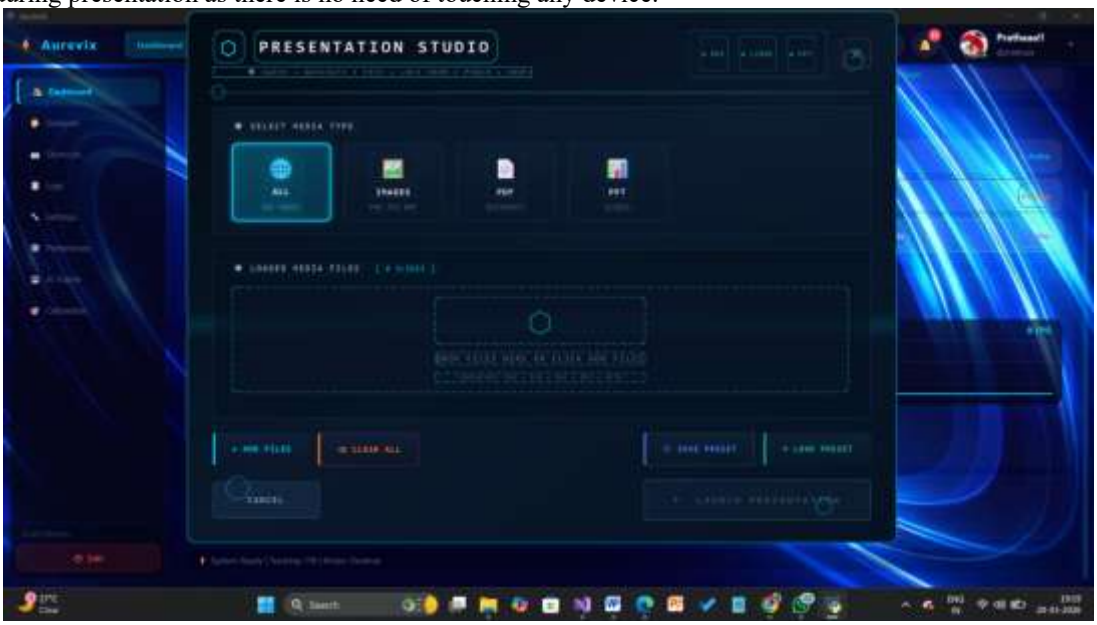

Fig6. Presentation Mode

5. **Training Module:** Training feature is added in the application where user can practice predefined gestures, which helps system understand user's movement patterns properly and improve overall interaction accuracy.

Fig7. Training Mode

6. **Calibration Module:** Calibration process adjusts the system according to user's hand size, movement range, and environment, which makes gesture detection more precise and reliable.

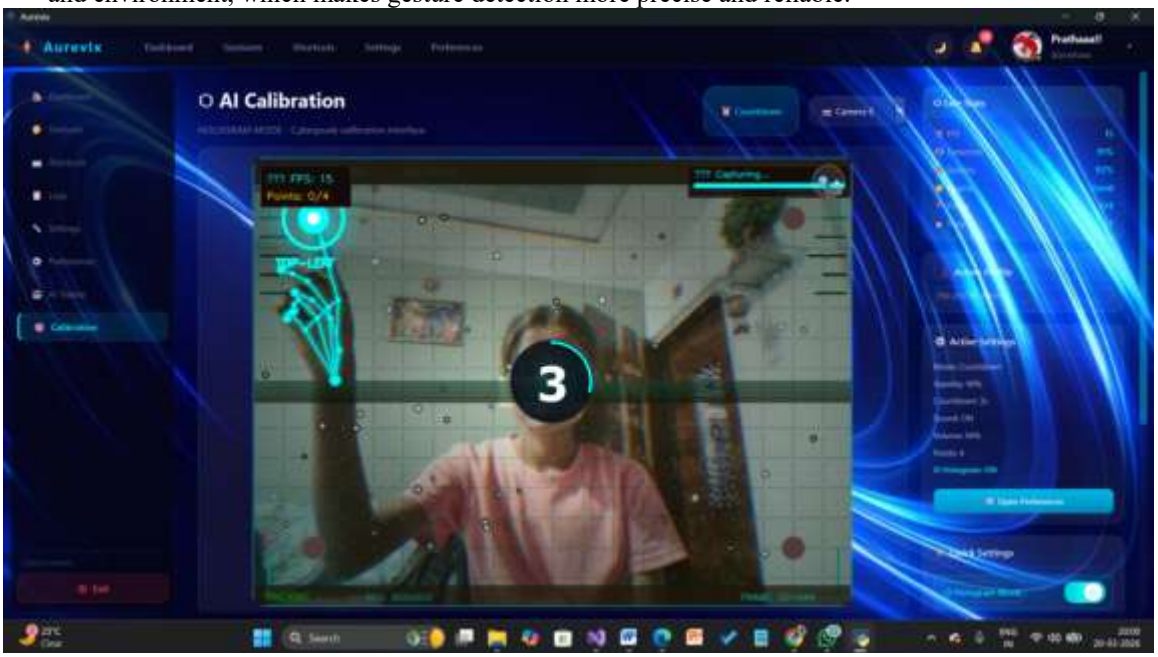

Fig8. Calibration

7. **Voice Input Module:** System performs specific actions by recognizing voice commands, which makes hand-free interaction possible and improves accessibility significantly.

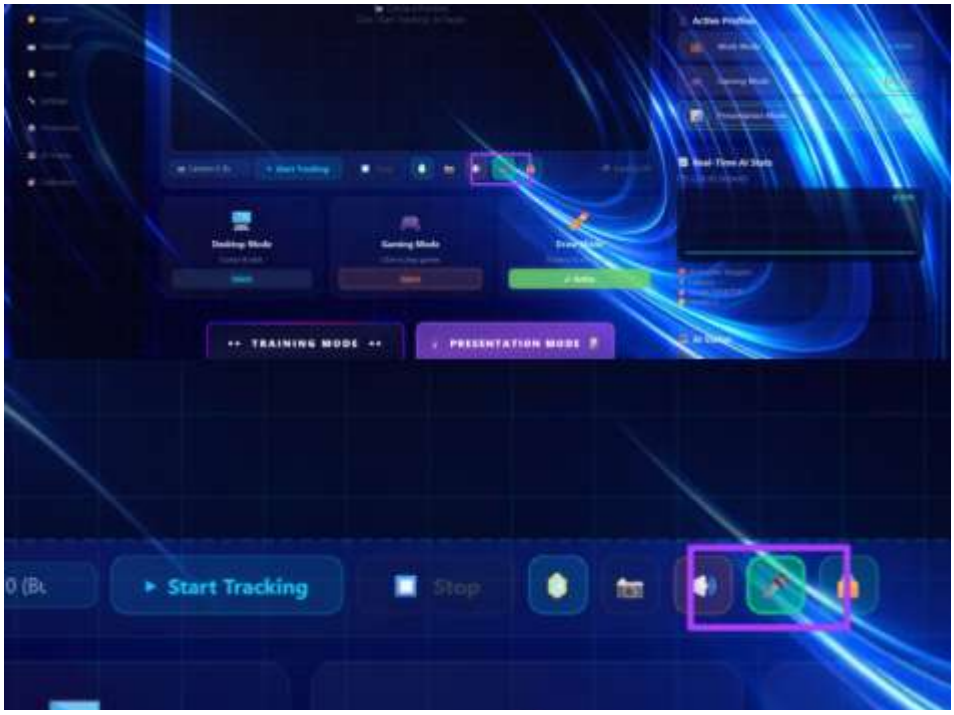

Fig9. Voice Input

Additional Support Features: Some more features are integrated in the system like gesture mapping and performance tracking that enhances usability and flexibility.

Result and Implementation- Aurevix is successfully implemented using a combination of real-time hand tracking algorithms, gesture recognition techniques and modular system control architecture. System is designed in such a way that it handles multiple interaction modes efficiently while maintaining real-time responsiveness.

Implementation Overview:

- System performs live hand detection and tracking through camera input, using continuous frame processing for accurate gesture capture.
- Detected hand movements and gestures are mapped with predefined actions, by which system-level like cursor control and navigation, and application-level commands like gaming, presentation are executed.
- Modular architecture is implemented in Aurevix where separate modules are designed for each mode, by which system performance stays optimized and improves scalability.
- Training module provides an environment where user can practice gestures, whereas calibration module adjusts the system according to user-specific parameters like hand size, motion range and positioning.
- Real-time processing pipeline is implemented in the system in which detection, recognition and execution stages are efficiently synchronized, ensuring minimal latency during interaction.
- Additional integration like voice input, gesture mapping, and performance monitoring makes system flexible and adaptive.

Results Achieved:-

- Real-time gesture recognition is successfully achieved by the system with smooth and responsive interaction across different modes.
- High usability is observed in Desktop mode and Presentation mode, where users can perform easily navigation and control without physical input device.
- Gaming mode and Draw mode significantly enhanced interactive experience, providing a more engaging and immersive environment.

- A noticeable improvement was seen in gesture detection accuracy by use of Training and Calibration modules, especially after repeated usage.
- System maintained a low-latency response, which is critical for real-time applications.
- Overall, Aurevix demonstrated an efficient, contactless and intuitive interaction model, which can be an effective and innovative of traditional input methods.

Limitations:-

Despite successful implementation and promising reults, some practical limitations are observed in Aurevix, which can create challenges during real-world deployment.

Identified Limitations:

1. **Accidental Gesture Detection:** System sometimes detects unintended or normal hand movements as valid gestures, especially in idle state or in presentation scenarios. This issue may affect user experience or may trigger unwanted actions.
2. **Lighting Dependency:** The performance of Aurevix largely depends on environmental lighting conditions. In case of low light, uneven lighting or excessive brightness, hand detection accuracy may reduce, which makes gesture recognition unreliable.
3. **Hardware Dependency:** To maintain smooth and real-time performance, system requires high-quality camera and sufficient processing power. In case of low-end devices latency may increase or may impact detection accuracy.
4. **Gesture Complexity Limitation:** Highly complex, fast, or visually similar gestures can be challenging to differentiate accurately. This can sometimes cause gesture misclassification, especially when gestures are not clearly distinguishable.
5. **User Adaptation Time:** There is an initial learning curve for new users to use the system effectively. Users require training time to understand the gesture, practice them, and adapt to the system.
6. **Environmental Constraints:** Detection process may interfere due to background clutter, multiple objects, or multiple hands in the frame, which may impact system accuracy and consistency.

Future Scope:

- Gesture recognition can be made more accurate by advanced AI and deep learning.
- Better adaptive learning can be implemented to make system more personalized.
- Mobile and cross-platform support can be added.
- Immersive interaction can be developed through AR/VR integration.
- Voice intelligent can be made more intelligent by natural language understanding.
- Gesture + voice hybrid system can be optimized.
- Integration with IoT devices and smart systems like TV, home automation is possible.
- Performance can be improved in different lighting conditions and complex backgrounds.

Conclusion:-

A smart gesture and voice-based human-computer interaction system named "Aurevix" is proposed in this paper. This system allows user to control computer with hand gestures and voice commands, without any need of physical device. System successfully provides multiple functionalities like virtual mouse control, drawing, gaming, presentation control and gesture training. Through real-time processing, smoothing techniques and adaptive learning, system ensures smooth and responsive interaction. Overall, Aurevix is a cost-effective, flexible and user-friendly solution that can be an efficient alternative of traditional input devices. This system is a strong step for modern touchless computing and can be expanded for advanced applications in future.

References:-

1.OpenCV Documentation. (2024). Open Source Computer Vision Library: Real-time Image Processing and Computer Vision. [Online]. Available: https://docs.opencv.org
2. MediaPipe by Google. (2024). On-device Real-time Hand Tracking and Landmark Detection Framework. [Online]. Available: https://developers.google.com/mediapipe
3.PyAutoGUI Documentation. (2024). Cross-platform GUI Automation for Mouse and Keyboard Control with Python. [Online]. Available: https://pyautogui.readthedocs.io

4.GeeksforGeeks. (2024). Gesture Controlled Virtual Mouse Implementation using Python and OpenCV. [Online]. Available: https://www.geeksforgeeks.org
5.GitHub Community. (2024). Open-source Repositories for Computer Vision, Hand Tracking, and Virtual Peripheral Simulation. [Online]. Available: https://github.com
6.Python Software Foundation. (2024). Python 3 Language Reference and Standard Library Documentation. [Online]. Available: https://docs.python.org/3
7.NumPy Documentation. (2024). Scientific Computing and Multi-dimensional Array Processing for Python. [Online]. Available: https://numpy.org/doc/