

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

Abstract:

1
2 Nowadays the demand of touchless human-computer interaction system is increasing, where traditional devices like
3 mouse and keyboard are not convenient in every situation. In this paper a Smart Gesture-Controlled System with
4 voice input named “Aurevix” is presented, which controls the computer with hand gestures and voice input. This
5 system uses webcam and microphone to get inputs from users and through MediaPipe, OpenCV and speech
6 processing technique detects gestures and voice commands. Aurevix supports multiple functionalities, like virtual
7 mouse control, drawing, gaming, presentation control and gesture training. This system works in real-time with
8 minimal delay and provides smooth interaction. Overall, this is cost-effective and user friendly solution, which
9 useful for modern touchless computing.

Key words:-

11 Gesture Recognition, Voice Recognition, Human-Computer Interaction (HCI), Multi-Modal Interaction, Machine Learning, Computer Vision,
12 Sensor-Based Control, Artificial Intelligence (AI), Speech Processing, Image Processing, Real-Time System, Automation, Hand Gesture
13 Detection, Natural User Interface (NUI), Deep Learning
14
15
16

.....
Introduction:-

17
18
19 For many years, the relation between human and a computer has been connected through some of the physical
20 devices. From the early days of mechanical trackballs to today’s advanced optical mouse, the way we interact with
21 computer has not changed yet; it has only improved slightly day by day. User still requires sitting at a desk, using a
22 flat surface and relying on physical hardware to control our machines. However, nowadays as technology and digital
23 environment has become more advanced and three-dimensional, these available traditional tools are starting to feel
24 outdated. Although there is lack of freedom, there are also some physical problems. The grip which is needed to
25 hold the mouse is lacking in some of the people. In addition, after the global pandemic, there is the need of clean and
26 touchless interfaces in places like, hospitals, laboratories, etc.
27

28 Using computer vision technology, Aurevix can detect and track 21 important points of our hand, also called as
29 landmarks, on the human hand in real time. These landmarks represent the joints and tips of the fingers. Instead of
30 simply moving a cursor, the system understands the movement and shape of the hand. It calculates the distance
31 between different fingers and parts of the palm using mathematical formula such as Euclidean distance. With this
32 information, the app converts visual hand movements into computer commands. This makes the interaction feel
33 natural and smooth, almost like communicating with the computer directly.

34 What makes Aurevix different from basic gesture tracking systems as it is inbuilt with Multi-Mode Architecture? In
35 the developmental phase of this app we came to know that different tasks require different levels of control and
36 sensitivity. A designer drawing on screen needs very precise movements, while a gamer needs fast and responsive
37 action. Because of this, we have designed four different operating modes that are been adjust according to the user
38 need thus making the system user efficient.

39 Desktop mode is designed for everyday tasks such as browsing the internet, opening files and general computer use.
40 It focuses on smooth cursor movement and prevents accidental clicks while working. Gaming mode focuses on speed
41 and performance. In this mode, the system reduces processing delays so that fast hand movements can quickly
42 translate into gaming actions. Draw mode is created for creative work such as digital art. It provides extremely stable

43 and accurate tracking so user draws or paint in the air with high precision. The multi control mode is designed for
44 long distance interaction. This allows the user to control voice assistant, presentations, volume, etc.

45 One major challenge when controlling a mouse using a webcam is a problem known as cursor jitter. Because
46 webcam can produce unwanted noise and human hands naturally make very small movements, the cursor may shake
47 or more slightly even when the user tries to keep their hand still. To solve such issues we have developed a recursive
48 smoothening algorithm. This algorithm works as a stabilizer. It filters out unwanted noise while keeping actual
49 movement of the hand. Due to which the cursor moves smoothly and steadily. So Aurevix is an app which focuses
50 on eliminating the physical devices and humans become the primary way to interact with computers.

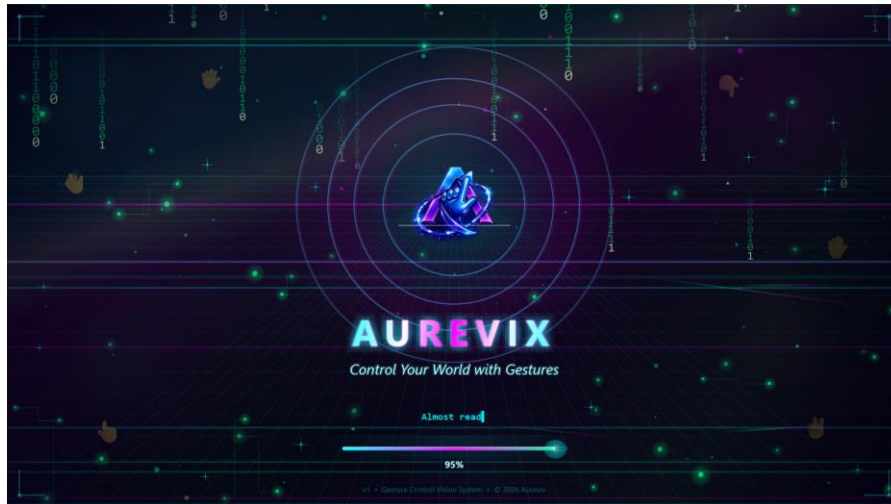


Fig.1: Splash Screen

Significance of the Study:-

55 This study demonstrates a highly accurate gesture-controlled virtual mouse system using computer vision
56 techniques, though it faces challenges in precise right-click operations and fine text selection [1]. The proposed
57 system integrates hand gestures with voice control to assist physically challenged users, but its performance is
58 affected by lighting conditions and system latency [2]. This research introduces a dual-mode gesture and voice-
59 controlled system with advanced automation features, yet its application is limited to specific operating systems
60 [3]. The system effectively replaces traditional input devices using webcam-based tracking, although it encounters
61 limitations in clicking accuracy and dragging precision [4]. The study achieves good accuracy in gesture recognition
62 using CNN and MediaPipe, but highlights the issue of high computational cost affecting real-time performance
63 [5]. This work presents a highly accurate gesture-controlled mouse with voice assistance, though further testing is
64 required under diverse environmental conditions [6]. The research focuses on enhancing system-level controls using
65 gesture recognition, but suggests improvements in accuracy and feature expansion for better usability [7]. The
66 system adapts well to varying lighting and finger positions, yet requires further development for scalability and web-
67 based implementation [8].

68 This study improves gesture stability using filtering techniques, but still faces challenges in performing complex
69 actions like dragging and right-clicking [9]. The research emphasizes inclusive and responsive interaction systems,
70 while recommending future enhancements such as depth sensing and adaptive learning [10].

Materials and Methods:-

73 **Proposed System-**Aurevix is architecture of high speed and vision based controller that is designed to
 74 reduce the dependence of physical Human-Computer Interaction (HCI) devices such as mouse and keyboard. We
 75 build this system to convert the raw hand gestures into mouse or keyboard action depending on the gesture, by
 76 shifting the complexity of spatial tracking to a dedicated software engine. Aurevix only require a standard 720p
 77 RGB camera.

78 **1) Functional Architecture and Data Pipeline**The internal logic of Aurevix is as followed:

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

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

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

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

The final cursor position is calculated using the following equation 108

$$P_{\text{final}} = (\alpha \cdot P_{\text{current}}) + ((1 - \alpha) \cdot P_{\text{average_past}}) \quad 109$$

Where, 110

P_{final} = Final cursor position 111

P_{current} = Current frame position 112

$P_{\text{average_past}}$ = Average position of previous frame 113

α = Factor of smoothing used to balance the current and previous position 114

115

116 2.Context-Aware Multi-Mode Logic

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

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

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

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

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

178 **VI) Voice Input:**Aurevix system is integrated with intelligent voice input feature, which allows user, the flexibility
179 to interact with gestures as well as voice commands. This feature acts as a virtual assistant, which understands user's
180 speech input and performs actions accordingly. As soon as the microphone is activated, system automatically

181 generates one context-aware greeting, which is on the basis of current time, like morning, afternoon or evening
182 greeting. This functionality makes the system more interactive and human-like, which improves the user experience.
183 At the backend of the system some commonly used voice commands are pre-defined, which user generally gives to
184 virtual assistant. Voice Command Processor module processes user's speech and converts them to meaningful
185 commands. Audio manager module efficiently handles the microphone input and captures smooth audio, whereas
186 Active AI Engine manages command execution and response generation. This feature supports real-time voice
187 processing, by which commands are quickly recognized and executed. The integration with gesture-based system
188 creates a hybrid interaction environment, where users can use gestures and voice according to their convenience.
189 This voice input functionality makes Aurevix system advanced, flexible and intelligent interaction platform, which
190 provides both natural communication and efficient control.

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

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

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

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

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

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

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

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

230 **8) System Advantages:**This architecture makes modular system design system highly scalable and maintainable.
231 Every module works independently, by which future upgrades and feature additions are easily possible. Real-time
232 processing, minimal latency and multi-mode support make Aurevix powerful and efficient gesture-based interaction
233 system.
234

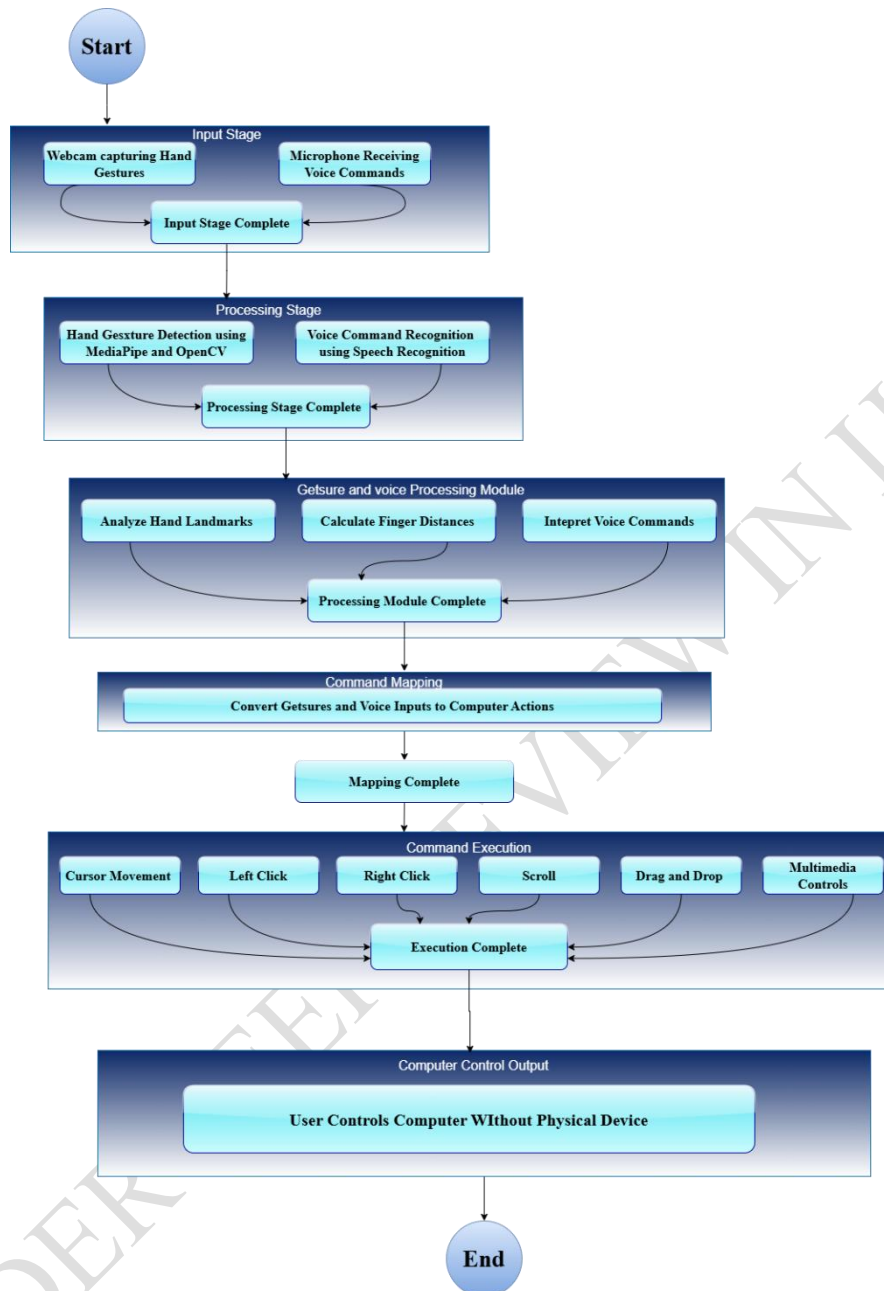


Fig.2: System Architecture

236

235

237

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

241 **1.Data Acquisition:**In this stage system collects input from the user. Webcam continuously captures video frames,
 242 in which user's hand gestures are detected. These frames are passed from real-time process.

243 At the same time, microphone captures voice input, which allows user to give commands. Both gesture and voice
 244 inputs are processed in parallel, enabling hybrid interaction.

245 **2.Image Pre-processing:**Captured video frames are not directly captured, but they are pre-processed. By using
 246 OpenCV frames are resized, flipped and optimized, so that detection becomes fast and accurate.For maintaining

247 noise reduction and frame consistency basic image processing techniques are applied, by which system performance
248 is improved.

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

252 **4.Gesture Recognition and Classification:**By using detected landmarks gestures are identified. Finger positions,
253 distances and angles are calculated, so that specific gestures can be detected.

- 254 • Cursor movements
- 255 • Click (left/right)
- 256 • Scroll
- 257 • Drag and Drop
- 258 • Drawing gestures
- 259 • Gaming action

260 For classification of gesture a rule-based logic is used, in which threshold and conditions are defined. To avoid
261 unintentional gestures filtering technique and delay validation are applied.

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

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

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

273 **8.Smoothing and Performance Optimization:**In gesture-based system cursor instability and jitter are one of the
274 common issues. To solve this smoothing algorithms are implemented, which stabilizes the input.Different mode has
275 different optimizations:

- 276 • In normal mode requires more smoothing.
- 277 • In gaming mode to reduce latency, smoothing is reduced.

278 This provides balanced performance.

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

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

285

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

289

290

291

292 **Core functionalities include:**

293
294
295

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

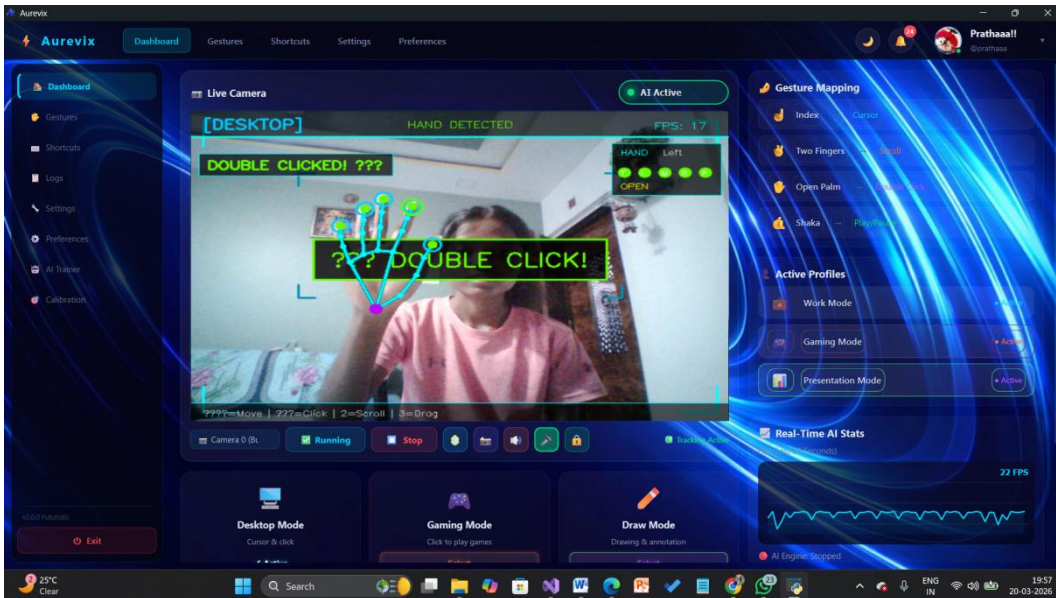


Fig3. Dashboard

296
297

298
299

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

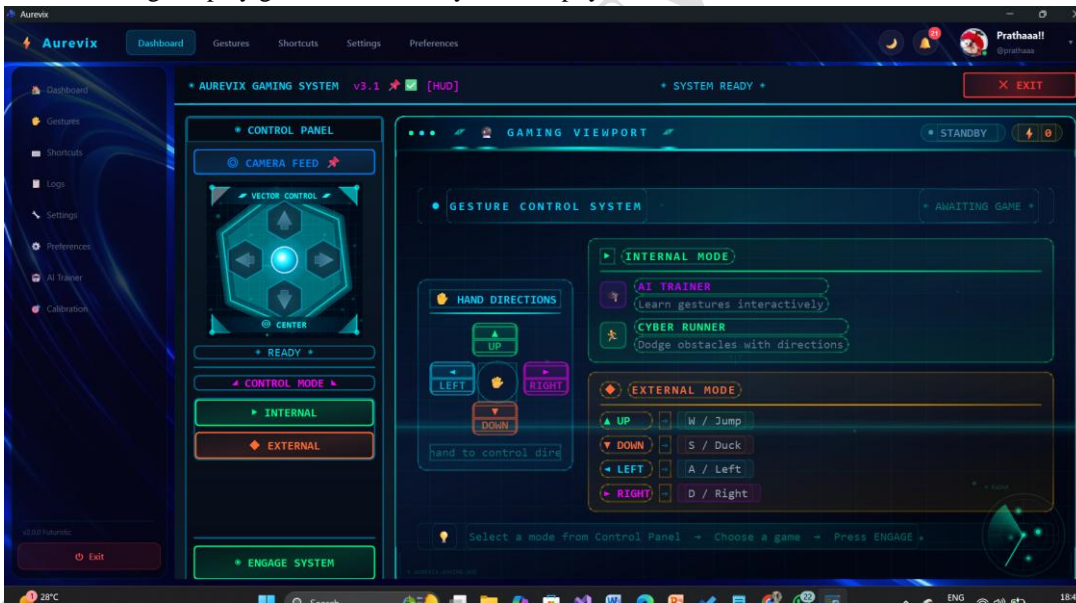


Fig4. Gaming System

300
301
302
303

304
305

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

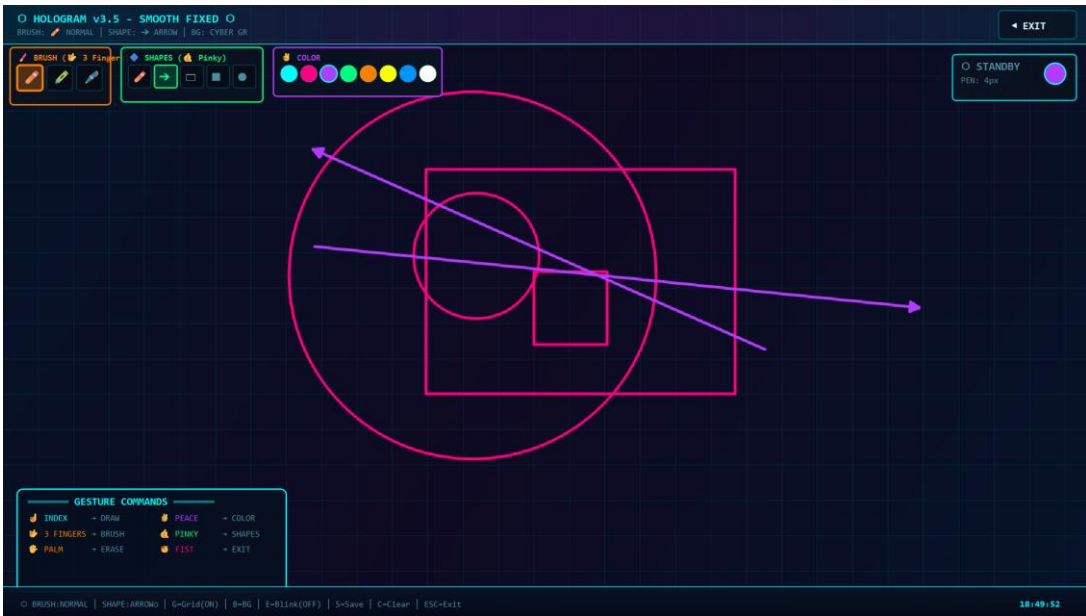


Fig5. Draw Mode

306
307

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

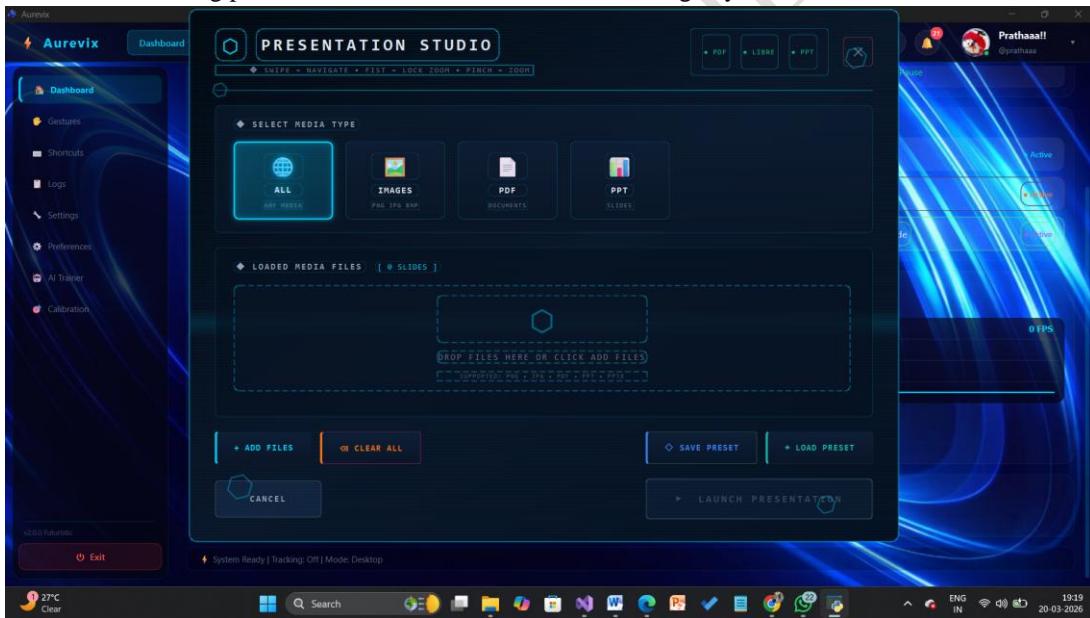


Fig6. Presentation Mode

310
311
312

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

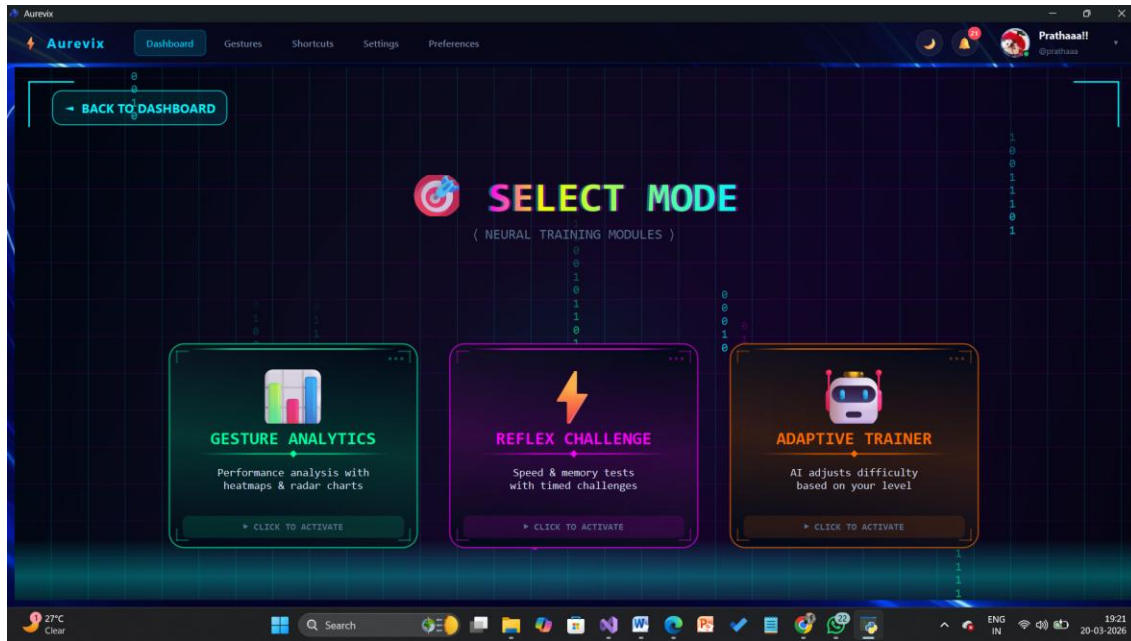


Fig7. Training Mode

316
317

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

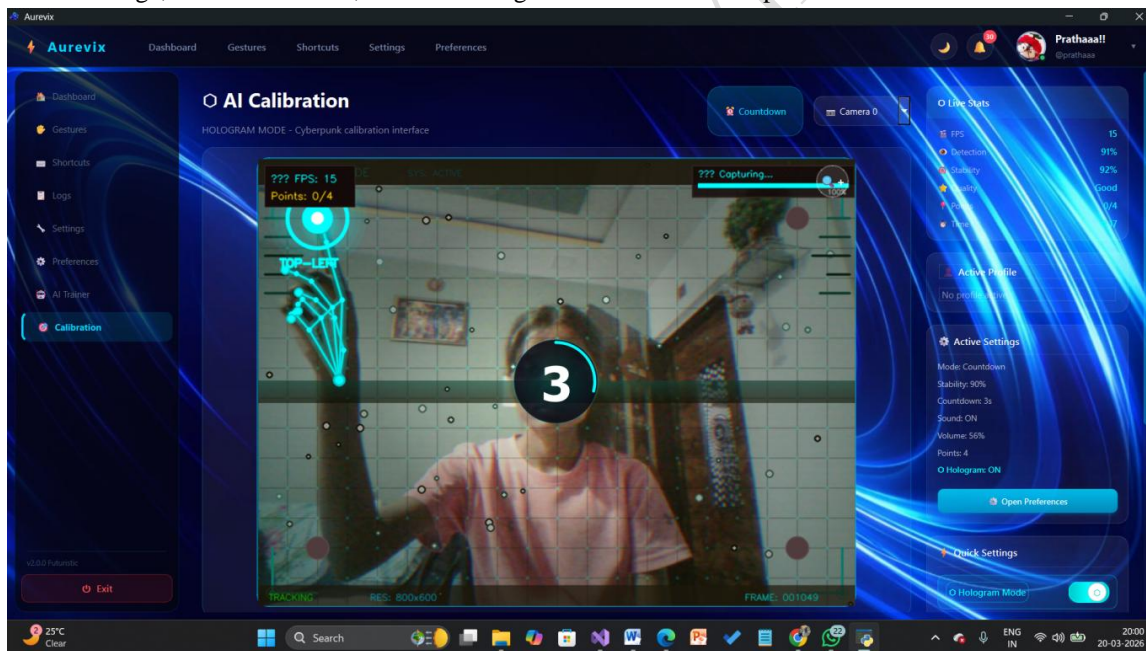


Fig8. Calibration

320
321
322
323
324

325

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

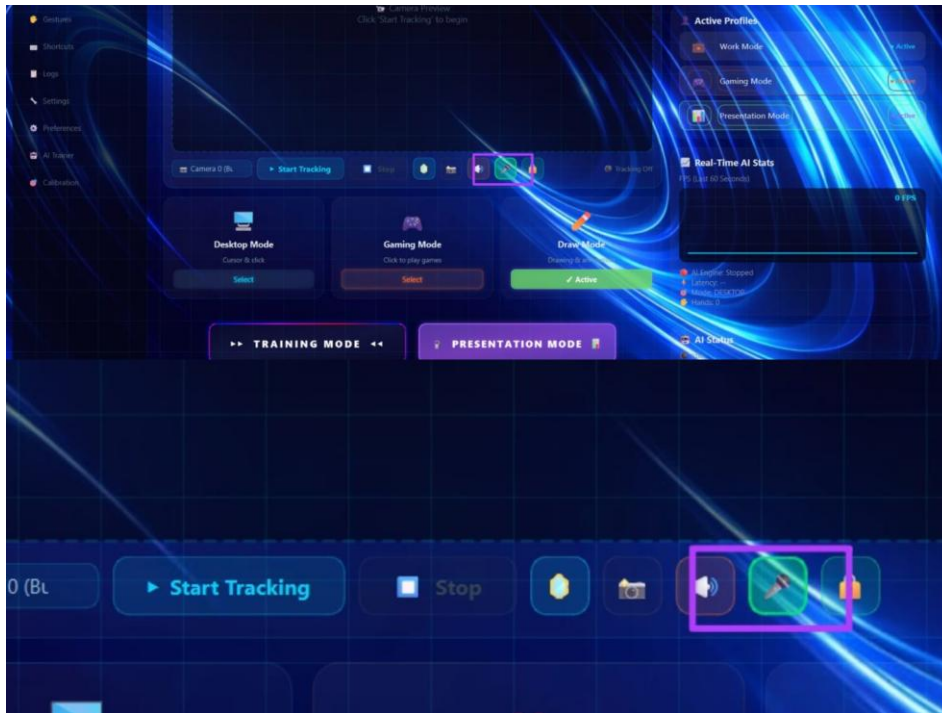


Fig9. Voice Input

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

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

336

337 **Implementation Overview:**

338 • System performs live hand detection and tracking through camera input, using continuous frame processing
 339 for accurate gesture capture.

340 • Detected hand movements and gestures are mapped with predefined actions, by which system-level like
 341 cursor control and navigation, and application-level commands like gaming, presentation are executed.

342 • Modular architecture is implemented in Aurevix where separate modules are designed for each mode, by
 343 which system performance stays optimized and improves scalability.

344 • Training module provides an environment where user can practice gestures, whereas calibration module
 345 adjusts the system according to user-specific parameters like hand size, motion range and positioning.

346 • Real-time processing pipeline is implemented in the system in which detection, recognition and execution
 347 stages are efficiently synchronized, ensuring minimal latency during interaction.

348 • Additional integration like voice input, gesture mapping, and performance monitoring makes system
 349 flexible and adaptive.

350

351

352 Results Achieved:

- 353 • Real-time gesture recognition is successfully achieved by the system with smooth and responsive
354 interaction across different modes.
- 355 • High usability is observed in Desktop mode and Presentation mode, where users can perform easily
356 navigation and control without physical input device.
- 357 • Gaming mode and Draw mode significantly enhanced interactive experience, providing a more engaging
358 and immersive environment.
- 359 • A noticeable improvement was seen in gesture detection accuracy by use of Training and Calibration
360 modules, especially after repeated usage.
- 361 • System maintained a low-latency response, which is critical for real-time applications.
- 362 • Overall, Aurevix demonstrated an efficient, contactless and intuitive interaction model, which can be an
363 effective and innovative of traditional input methods.

364

365 **Limitations:** Despite successful implementation and promising results, some practical limitations are observed in
366 Aurevix, which can create challenges during real-world deployment.

367 Identified Limitations:

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

384 Future Scope:

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

393

394

395 Conclusion:

396 A smart gesture and voice-based human-computer interaction system named "Aurevix" is proposed in this paper.
397 This system allows user to control computer with hand gestures and voice commands, without any need of physical
398 device.

399 System successfully provides multiple functionalities like virtual mouse control, drawing, gaming, presentation
400 control and gesture training. Through real-time processing, smoothing techniques and adaptive learning, system
401 ensures smooth and responsive interaction.

402 Overall, Aurevix is a cost-effective, flexible and user-friendly solution that can be an efficient alternative of
403 traditional input devices. This system is a strong step for modern touchless computing and can be expanded for
404 advanced applications in future.

405

406 References:

407 1.OpenCV Documentation. (2024). Open Source Computer Vision Library: Real-time Image Processing and
408 Computer Vision. [Online]. Available: <https://docs.opencv.org>

409 2. MediaPipe by Google. (2024). On-device Real-time Hand Tracking and Landmark Detection Framework.
410 [Online]. Available: <https://developers.google.com/mediapipe>

411 3.PyAutoGUI Documentation. (2024). Cross-platform GUI Automation for Mouse and Keyboard Control with
412 Python. [Online]. Available: <https://pyautogui.readthedocs.io>

413 4.GeeksforGeeks. (2024). Gesture Controlled Virtual Mouse Implementation using Python and OpenCV. [Online].
414 Available: <https://www.geeksforgeeks.org>

415 5.GitHub Community. (2024). Open-source Repositories for Computer Vision, Hand Tracking, and Virtual
416 Peripheral Simulation. [Online]. Available: <https://github.com>

417 6.Python Software Foundation. (2024). Python 3 Language Reference and Standard Library Documentation.
418 [Online]. Available: <https://docs.python.org/3>

419 7.NumPy Documentation. (2024). Scientific Computing and Multi-dimensional Array Processing for Python.
420 [Online]. Available: <https://numpy.org/doc/>

421