MSI Journal of
AI and Technology

Volume: 2, Issue: 3 (July-September 2026)
Published by *MSI Publishers*
ISSN 3107-6181 (Online)

Recived: 15-05-2026
Accepted: 02-07-2026
Published: 07-07-2026

DIY AI Smart Glass: Design, Development, and Evaluation of an Indigenous Multimodal Wearable Assistant

Authors Details

Ramya Dharmesh Kumar Patel[1*]
Khyati World School, Ahmedabad, India.

Dr. Priyam Parikh[2]
Assistant Professor, School of Design, Anant National University, Ahmedabad Gujarat.

ABSTRACT: The rapid advancement of multimodal artificial intelligence has accelerated the development of wearable smart assistants capable of integrating vision, speech, and contextual reasoning. However, commercially available AI smart glasses remain expensive, proprietary, and difficult to customize, limiting their accessibility for education, research, and assistive technology development, particularly in resource-constrained environments. This study presents DIY AI Smart Glass, an indigenous low-cost multimodal wearable assistant designed to provide an open-source alternative for real-time human–computer interaction. The primary aim of this research is to develop and evaluate an affordable wearable AI platform capable of understanding spoken language, interpreting visual information, and generating contextual voice responses through an integrated multimodal framework. The proposed system employs an ESP32-S3 AI camera module equipped with an OV3660 camera, PDM microphone, and inbuilt speaker, while computationally intensive multimodal inference is performed on a host computer using a Python-based architecture integrated with a cloud-based large multimodal language model. The methodology incorporates wireless acquisition of audio and visual data, wake-word activation, speech understanding, scene interpretation, contextual reasoning, and real-time speech synthesis, enabling natural hands-free interaction through a single wearable device.

DOI: *10.5281/zenodo.21233518*

Unlike conventional wearable vision systems that primarily perform object detection, the proposed framework supports both visual scene understanding and general conversational assistance within a unified architecture. Experimental validation was conducted across representative visual and non-visual interaction scenarios to evaluate functionality, response generation, and user experience. The developed prototype demonstrates reliable multimodal interaction while maintaining a significantly lower hardware cost than commercial smart glasses, thereby providing an accessible platform for assistive technologies, educational robotics, wearable artificial intelligence research, and rapid prototyping of next-generation human-centred intelligent systems.

Keywords: *Multimodal Artificial Intelligence; Wearable Smart Glasses; Human–Computer Interaction; ESP32-S3; Large Language Models (LLMs).*

1. Background

Wearable computing has experienced remarkable growth over the past decade, evolving from simple activity trackers into intelligent devices capable of perceiving, interpreting, and responding to their surrounding environment [1]. Recent advances in artificial intelligence, embedded electronics, computer vision, wireless communication, and large language models have accelerated the development of wearable systems that enable natural and intuitive human–computer interaction. Among these technologies, smart glasses have emerged as one of the most promising wearable platforms because they provide users with hands-free access to visual information, voice interaction, augmented intelligence, and contextual assistance while maintaining mobility [2]. These capabilities have expanded the application of smart glasses across healthcare, industrial maintenance, education, navigation, assistive technologies, and immersive human–machine interaction [3].

Modern commercial smart glasses integrate cameras, microphones, speakers, wireless connectivity, and cloud-based artificial intelligence to assist users in performing everyday tasks [4]. Users can interact with these systems using natural speech, receive spoken responses, identify surrounding objects, obtain scene descriptions, translate text, perform internet searches, and access contextual information without requiring conventional handheld devices. Such multimodal interaction significantly enhances accessibility and convenience while reducing cognitive and physical effort [5]. The emergence of large

 DOI: *10.5281/zenodo.21233518*

multimodal language models has further transformed wearable computing by enabling systems to simultaneously understand images, speech, and textual information, allowing more intelligent reasoning and conversational interaction than conventional object detection or voice-assistant technologies [5], [6].

Despite these technological advances, commercially available AI smart glasses remain inaccessible to many researchers, educational institutions, students, and developers due to their high cost, proprietary hardware architecture, and limited opportunities for hardware customization. Most commercial devices operate within closed ecosystems where both hardware and software modifications are restricted. Consequently, researchers often face significant challenges in evaluating novel interaction paradigms, integrating custom sensors, experimenting with alternative artificial intelligence algorithms, or developing application-specific wearable systems. These limitations are particularly evident in developing countries where budget constraints frequently restrict access to advanced wearable computing platforms. As a result, the development of affordable, open, and customizable wearable AI systems has become an important research direction for promoting innovation and democratizing access to intelligent wearable technologies [6], [7], [8].

Several research efforts have investigated wearable assistive systems based on computer vision, embedded processors, and machine learning algorithms. Existing prototypes have demonstrated applications including object recognition, facial recognition, obstacle detection, navigation assistance, optical character recognition, text-to-speech conversion, and environmental awareness for visually impaired individuals [2], [3], [9]. More recent developments have incorporated deep learning techniques to improve recognition accuracy and contextual understanding. However, many of these systems remain application-specific, relying on predefined object classes or fixed machine learning models that cannot easily respond to arbitrary user questions or engage in natural conversations. Consequently, their interaction capabilities remain limited when compared with modern multimodal artificial intelligence systems [7], [8].

The introduction of cloud-based large language models has substantially expanded the capabilities of wearable intelligent systems. Instead of restricting users to predefined commands, multimodal foundation models can interpret natural language, analyse visual

 DOI: *10.5281/zenodo.21233518*

scenes, answer contextual questions, explain surrounding environments, provide educational information, and engage in conversational dialogue [10], [11], [12], [13]. The integration of vision-language models into wearable devices therefore represents a significant advancement toward more natural human–computer interaction. Nevertheless, deploying these capabilities directly on embedded wearable hardware remains computationally challenging because advanced multimodal models require processing resources that exceed the capabilities of low-power microcontrollers. Hybrid edge–cloud architectures therefore provide a practical solution by combining low-cost embedded sensing devices with cloud-based artificial intelligence services [1], [4].

The ESP32-S3 microcontroller has recently emerged as a highly attractive embedded platform for intelligent Internet of Things and wearable applications because it integrates wireless communication, sufficient computational resources, low power consumption, and support for camera, microphone, and speaker interfaces within a compact and economical hardware platform. The availability of camera modules equipped with integrated microphones and speakers enables the rapid development of multimodal wearable devices without requiring complex custom electronic hardware. Furthermore, open-source development environments, readily available software libraries, and extensive community support have made the ESP32-S3 an attractive platform for educational research, rapid prototyping, and proof-of-concept demonstrations. However, relatively few studies have explored the integration of ESP32-S3-based wearable hardware with contemporary multimodal large language models for creating conversational AI smart glasses capable of both visual perception and general-purpose knowledge assistance [1], [4], [8].

The present work addresses this research gap through the design and development of DIY AI Smart Glass, an indigenous low-cost multimodal wearable assistant intended to provide an affordable and customizable alternative to commercial AI smart glasses. Rather than replicating proprietary commercial products, the proposed system establishes an open hardware and software platform that can be modified, extended, and adapted for diverse research and educational applications. The developed prototype integrates an ESP32-S3 AI camera module containing an OV3660 camera, PDM microphone, and inbuilt speaker with a Python-based processing framework communicating wirelessly through a Wi-Fi network [14]. Computationally intensive multimodal reasoning is performed using a cloud-hosted

 DOI: *10.5281/zenodo.21233518*

large multimodal language model, enabling the wearable device to interpret spoken commands, analyse visual scenes when required, answer general knowledge questions, and generate natural spoken responses through its integrated speaker. The architecture combines embedded sensing with cloud intelligence while maintaining a lightweight wearable form factor and significantly reducing overall system cost [15].

Unlike many existing wearable vision systems that primarily focus on object detection or scene recognition, the proposed framework introduces an adaptive interaction strategy that first determines user intent before deciding whether visual information is required [16]. General conversational questions are answered directly through multimodal language understanding, whereas visual questions trigger image acquisition and contextual scene analysis. This approach reduces unnecessary image transmission, improves system efficiency, decreases network usage, and creates a more natural conversational experience. The incorporation of wake-word activation, contextual reasoning, multimodal understanding, and real-time voice feedback further enhances usability by allowing users to interact with the system in an intuitive hands-free manner [17].

The primary objective of this research is to design, develop, and experimentally evaluate a low-cost indigenous AI smart glass capable of multimodal interaction using commercially available embedded hardware and cloud-based artificial intelligence services. The study further aims to demonstrate that sophisticated wearable AI assistance can be achieved using affordable open-source hardware without requiring specialized custo [18] electronics or expensive proprietary platforms [14]. The proposed framework is expected to contribute to the growing field of wearable artificial intelligence by providing a scalable and reproducible platform suitable for assistive technologies, educational laboratories, robotics research, human–computer interaction, rapid prototyping, and future investigations into embedded multimodal intelligence. Through the integration of low-cost hardware, wireless communication, multimodal artificial intelligence, and conversational interaction, the proposed DIY AI Smart Glass establishes a practical foundation for the next generation of affordable intelligent wearable assistants.

 DOI: *10.5281/zenodo.21233518*

2. Literature Survey

Table 1: Literature Survey with Hardware, Method and Research Gap

Ref.	Application	Hardware Used	Methods / AI Techniques	Major Findings	Research Gap
[4], [8]	AI-enabled smart glasses for vision assistance	Meta Smart Glasses, camera, microphone, speaker	Large Language Models (LLMs), Computer Vision, Speech Interaction	Demonstrated conversational AI integrated with wearable smart glasses for assistive applications.	Commercial proprietary platform, expensive, closed ecosystem, limited hardware customization.
[4]	AI smart glasses in digital healthcare	Various commercial smart glasses	Systematic review, AI, multimodal sensing	Summarized healthcare applications including diagnosis, rehabilitation, and remote monitoring.	No implementation framework or low-cost prototype presented.
[6]	Large Language Models for smart glasses	Commercial wearable platforms	Multimodal LLMs, Vision-Language Models	Proposed future direction for ubiquitous AI using wearable devices.	Conceptual framework without indigenous hardware implementation.
[9]	Mobility assistance	XR Smart Glasses	YOLOv8, Computer Vision	Achieved real-time obstacle detection and environmental awareness.	Limited to object detection; lacks conversational multimodal interaction.
[3]	Spatial awareness	Smart glasses with camera	Deep Learning, Computer Vision	Improved environmental perception for assistive navigation.	Does not support natural language understanding or general-purpose AI conversations.
[5]	Smart glasses for blind, deaf and mute users	Arduino, Camera, IoT modules	Machine Learning, IoT, Speech Processing	Demonstrated affordable assistive wearable system.	Traditional ML with predefined tasks; no multimodal reasoning.

DOI: *10.5281/zenodo.21233518*

[5], [7]	Blind assistance	Embedded camera system	Deep Learning, CNN	Improved object recognition accuracy.	Focused only on visual perception without conversational AI.
[7]	LidSonic assistive glasses	Arduino, ultrasonic sensors, smartphone	Green Machine Learning, Mobile App	Low-power wearable navigation system.	Relies on conventional ML and navigation; lacks scene understanding and LLM integration.
[2]	Facial recognition	Raspberry Pi, Camera	Face Recognition, IoT	Recognized known individuals for visually impaired users.	Restricted to face recognition without contextual reasoning.
[1]	Visual assistance	Embedded wearable platform	Deep Learning, Object Detection	Improved accessibility using AI vision techniques.	Application-specific implementation without multimodal conversational intelligence.

Table 1 summarizes the major developments in AI-enabled smart glasses and wearable assistive systems by comparing their applications, hardware platforms, computational methods, key contributions, and research gaps. The reviewed studies show that existing systems commonly focus on object detection, facial recognition, obstacle avoidance, spatial awareness, healthcare support, or vision assistance using commercial smart glasses, Arduino/Raspberry Pi platforms, XR devices, IoT modules, and deep learning methods. However, most prior works remain application-specific, proprietary, expensive, or limited to predefined visual tasks. This highlights the need for a low-cost indigenous multimodal wearable assistant capable of both visual understanding and general conversational interaction.

3. Problem Statement, Objectives and Methodology

Commercial AI smart glasses demonstrate the potential of hands-free visual assistance, speech interaction, and contextual computing; however, these devices are commonly expensive, proprietary, and difficult to customize for academic research, assistive

 DOI: *10.5281/zenodo.21233518*

technology development, and low-resource educational prototyping. Existing research prototypes frequently focus on isolated functions such as object detection, facial recognition, obstacle avoidance, or navigation support, while offering limited integration of camera-based perception, microphone-based speech input, speaker-based feedback, and general-purpose multimodal reasoning within a single low-cost wearable system [1]-[10]. Therefore, there is a clear need for an indigenous, affordable, and reproducible AI smart glass platform that combines embedded sensing with cloud-based multimodal intelligence to support both visual scene understanding and non-visual conversational assistance through a natural voice-driven interface. Following are the objectives.

1. To design and develop a low-cost indigenous AI smart glass prototype using an ESP32-S3 camera module with integrated camera, microphone, speaker, and Wi-Fi communication.
2. To implement a multimodal interaction pipeline capable of wake-word activation, speech understanding, visual scene analysis, general question answering, and spoken response generation.
3. To evaluate the functional performance, usability, response reliability, and affordability of the proposed wearable assistant across representative visual and non-visual interaction scenarios.

Table 2. Objectives versus methodology for the proposed DIY AI Smart Glass

Objective	**Methodology / Technical Approach**	**Hardware / Software Used**	**Evaluation Indicators**	**Relevant Citations**
Objective 1: Develop low-cost indigenous AI smart glass hardware.	A wearable prototype will be developed using an ESP32-S3 AI camera module. The device will integrate image acquisition, audio capture, speaker	ESP32-S3 AI camera module, OV3660 camera, PDM microphone, inbuilt speaker, Wi-Fi network, 3D-printed or DIY wearable frame,	Hardware cost, successful camera capture, successful microphone recording, successful speaker playback,	[2], [3], [9], [19]

 DOI: *10.5281/zenodo.21233518*

Objective	Methodology / Technical Approach	Hardware / Software Used	Evaluation Indicators	Relevant Citations
	output, and Wi-Fi communication. The design will focus on affordability, reproducibility, compactness, and ease of assembly.	Arduino IDE.	device portability, and reproducibility.	
Objective 2: Implement multimodal AI interaction.	A Python-based edge-cloud pipeline will be implemented to receive audio and image data from the ESP32-S3. The system will use wake-word activation, speech understanding, conditional image capture, multimodal reasoning, and TTS-based voice feedback through the ESP32 speaker.	Python, Google Gemini multimodal API, pyttsx3, pydub, FFmpeg, HTTP communication, ESP32 camera/audio endpoints.	Wake-word response, correct interpretation of spoken queries, visual scene description quality, general question-answering success, and successful audio response generation.	[14], [15], [16], [17]
Objective 3: Evaluate functionality, usability, and affordability.	The prototype will be tested across representative visual and non-visual scenarios, including object identification,	Prototype device, predefined test scenarios, response logs, user observation, Likert-scale	Response latency, successful task completion rate, speech output success rate, perceived	[1], [7], [18]

DOI: *10.5281/zenodo.21233518*

Objective	Methodology / Technical Approach	Hardware / Software Used	Evaluation Indicators	Relevant Citations
	scene description, general knowledge questions, and hands-free interaction. Qualitative feedback and quantitative system measures will be collected.	feedback, latency measurement, and cost analysis.	usability, interaction reliability, and total system cost.	

Table 2 presents the relationship between the proposed objectives and the corresponding methodological approach. The methodology is aligned with prior smart-glass and assistive wearable studies while addressing their limitations through low-cost hardware, multimodal reasoning, and conversational interaction.

4. Scientific Diagram, Flowchart and Developed Product

Figure 1 presents the overall scientific framework of the proposed DIY AI Smart Glass, including the system architecture, operational flow, and developed 3D-printed wearable prototype. The architecture is divided into three functional layers: the AI smart glass edge unit, the local processing layer, and the cloud-based multimodal artificial intelligence layer. The edge unit integrates an ESP32-S3 AI camera module with an OV3660 camera, PDM microphone, inbuilt speaker, wake-word input mechanism, and a customized 3D-printed eyeglass frame. The device communicates wirelessly with the local processing system through Wi-Fi, enabling audio capture, speech-to-text conversion, query classification, and conditional image acquisition. A key feature of the proposed framework is its adaptive decision logic, where the system first determines whether visual information is required. If the user asks a general question, only the text query is forwarded to the large language model; however, if the user asks a visual question, both the captured image and text query are transmitted for multimodal reasoning. The cloud AI layer performs text understanding, image understanding, contextual reasoning, response generation, and text-to-speech

 DOI: *10.5281/zenodo.21233518*

synthesis. The flowchart further illustrates the sequential operation from wake-word detection and audio capture to response playback through the onboard speaker. The developed product images confirm the practical realization of the proposed architecture in a compact wearable form factor. Overall, Figure 1 demonstrates how low-cost embedded hardware, local processing, and cloud-based multimodal intelligence are integrated to create an indigenous wearable assistant capable of hands-free visual and conversational interaction.

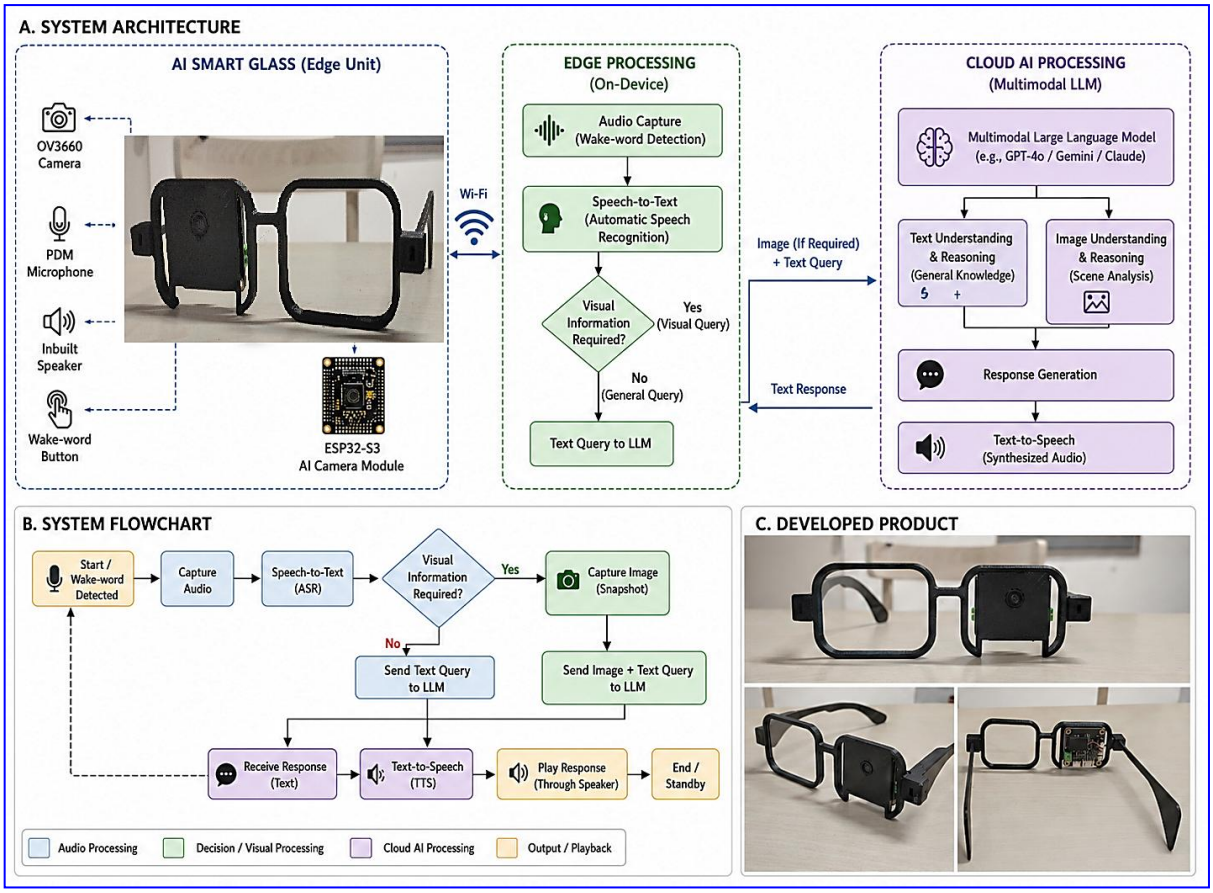

Figure 1: Scientific Diagram of the Product

The proposed DIY AI Smart Glass was developed as a hybrid edge–cloud multimodal wearable assistant using an ESP32-S3 AI camera module, a 3D-printed eyeglass frame, Arduino firmware, and a Python-based gateway application [20], [21], [22]. The hardware unit consisted of an OV3660 camera, PDM microphone, inbuilt speaker, ESP32-S3 microcontroller, and Wi-Fi interface mounted on a customized 3D-printed spectacle frame. In the first stage, the ESP32-S3 board was configured in Arduino IDE using the ESP32-S3 Dev Module settings, followed by basic onboard LED testing to confirm successful

 DOI: *10.5281/zenodo.21233518*

firmware upload and GPIO operation. The OV3660 camera was then initialized using the ESP32 camera driver and configured to operate through a Wi-Fi-based camera server. This enabled image acquisition through an HTTP /capture endpoint, allowing the Python program to request a JPEG image only when visual information was required.

The microphone and speaker subsystems were developed separately and then integrated into the final firmware. The inbuilt PDM microphone was initialized using the ESP_I2S library with GPIO38 as the PDM clock and GPIO39 as the data input. Audio was recorded as a WAV buffer using the ESP32-S3 and exposed through an HTTP /record endpoint on port 82. The inbuilt speaker was configured through I2S using GPIO45 as BCLK, GPIO46 as LRCK, and GPIO42 as DOUT. A separate /play endpoint was created to receive WAV audio from the Python gateway and play it through the onboard speaker. Thus, the Arduino firmware provided three core services: image capture, audio recording, and speaker playback [23], [24], [25].

On the software side, Python acted as the local gateway between the wearable device and the cloud multimodal AI model. The Python program requested recorded audio from the ESP32-S3 microphone, sent the audio to Gemini for wake-word and intent analysis, and determined whether the user said “Hey Krishna.” If the wake word was not detected, the system returned to standby mode. If the wake word was detected, the model classified the query as either general or visual. General questions were answered directly using audio-only reasoning, while visual questions triggered image capture from the ESP32-S3 camera. The selected input data were then submitted to Gemini for multimodal reasoning. To improve reliability, multiple Gemini models were included as fallback options.

After Gemini generated a short response, Python converted the text into speech using pyttsx3. Since the ESP32-S3 speaker required a compatible WAV format, the generated speech was converted using pydub into 16 kHz, mono, 16-bit PCM WAV audio. This WAV file was transmitted to the ESP32-S3 /play endpoint and played through the inbuilt speaker. After each response, the assistant prompted the user by saying, “User, what can I do for you?” This complete workflow enabled a hands-free interaction cycle involving wake-word activation, speech understanding, conditional visual sensing, multimodal reasoning, speech synthesis, and embedded speaker output.

 DOI: *10.5281/zenodo.21233518*

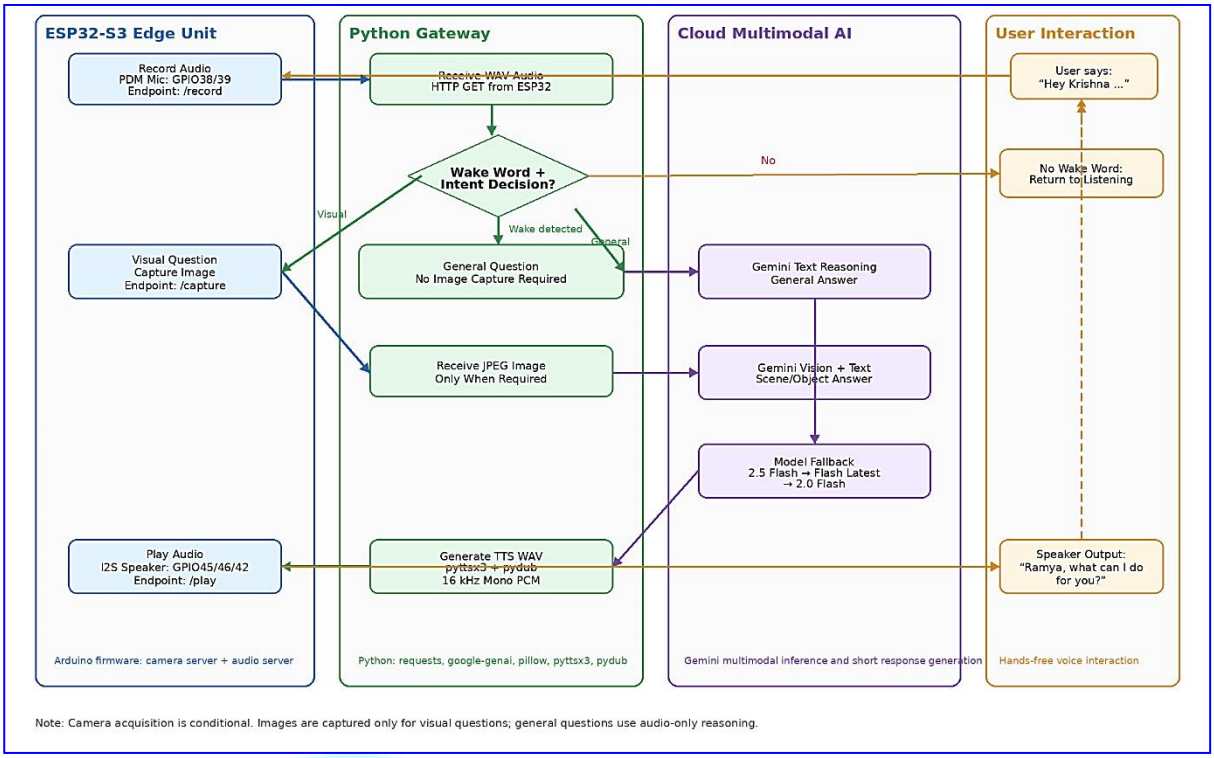

Figure 2: Complete Flowchart of the System

Figure 3 shows the developed 3D-printed indigenous AI Smart Glass prototype fabricated as a low-cost wearable platform for multimodal interaction. The frame was designed to resemble a conventional eyeglass structure while providing sufficient mechanical support for mounting the ESP32-S3 AI camera module. The design was first modelled in CAD with provisions for the front-facing camera opening, board placement, side housing, wire routing, and wearable temple arms. The enclosure was designed to hold the electronic module securely near the eye region without obstructing the user's field of view. After finalizing the geometry, the parts were fabricated using fused deposition modelling 3D printing with black polymer filament, providing a lightweight, rigid, and easily reproducible structure. The ESP32-S3 camera board was mounted inside the printed enclosure using screw-based support points, while the camera lens was aligned with the front aperture to enable visual capture. The onboard microphone and speaker remained accessible through the casing, allowing audio input and output during interaction. The temple arms were extended to improve stability during wearing, and the modular design allows easy removal or replacement of the electronic board. This prototype demonstrates how affordable additive manufacturing and embedded electronics can be combined to create an indigenous wearable AI assistant suitable for research, education, and rapid prototyping.

DOI: *10.5281/zenodo.21233518*

Figure 3: 3D Printed Indigenous AI Smart Glass

Table 3. Software Libraries Used in the Development of the DIY AI Smart Glass

Platform	Library / Module	Purpose	Use in the Proposed System
Arduino / ESP32-S3	WiFi.h	Wireless communication	Connected the ESP32-S3 module to the local Wi-Fi network for camera, microphone, and speaker communication.
Arduino / ESP32-S3	esp_camera.h	Camera control	Initialized the OV3660 camera and enabled image capture through the /capture HTTP endpoint.
Arduino / ESP32-S3	ESP_I2S.h	Audio input/output	Configured the PDM microphone for audio recording and I2S speaker for audio playback.
Arduino / ESP32-S3	esp_http_server.h	HTTP server creation	Created /record and /play endpoints for audio recording and speaker playback.
Arduino / ESP32-S3	SPI.h	Peripheral communication support	Supported embedded communication functions used during audio and peripheral handling.

DOI: *10.5281/zenodo.21233518*

Python	requests	HTTP communication	Sent and received data between Python and ESP32-S3 through /capture, /record, and /play.
Python	google-genai	Gemini API integration	Sent audio, image, and text prompts to Gemini for multimodal reasoning and response generation.
Python	Pillow	Image handling	Converted JPEG images received from ESP32-S3 into a format suitable for Gemini processing.
Python	pyttsx3	Text-to-speech generation	Converted Gemini text responses into spoken audio files.
Python	pydub	Audio conversion	Converted generated speech into 16 kHz, mono, 16-bit PCM WAV format compatible with ESP32-S3 speaker playback.
Python	os	File handling	Managed temporary WAV files generated during text-to-speech conversion.
Python	uuid	Unique file naming	Generated unique filenames to avoid repeated audio playback conflicts.
Python	time	Delay and timing control	Controlled listening intervals, retry delays, and system response timing.
Python	json	Structured response parsing	Parsed Gemini responses for wake-word detection, intent classification, and camera-use decision.
Python	re	Text extraction	Extracted JSON-like responses when Gemini returned additional text around structured output.

DOI: *10.5281/zenodo.21233518*

Table 3 summarizes the major Arduino and Python libraries used to implement the proposed DIY AI Smart Glass. The Arduino-side libraries enabled low-level embedded functions such as Wi-Fi connectivity, camera initialization, microphone recording, speaker playback, and HTTP endpoint creation. These libraries allowed the ESP32-S3 module to operate as an edge sensing and response unit. The Python-side libraries formed the local gateway layer, managing communication with the ESP32-S3, processing image and audio data, integrating Gemini multimodal reasoning, generating speech, and converting audio into an ESP32-compatible WAV format. Together, these libraries enabled the complete multimodal pipeline involving wake-word activation, conditional visual sensing, cloud-based reasoning, and spoken response output.

5. Algorithm

Algorithm 1 describes the wake-word-based multimodal interaction strategy implemented in the proposed DIY AI Smart Glass. The system begins by initializing the ESP32-S3 hardware, including the OV3660 camera, PDM microphone, I2S speaker, Wi-Fi interface, and HTTP endpoints for audio recording, image capture, and speaker playback. The Python gateway then communicates with the wearable device and plays an initial welcome message to the user. During operation, the ESP32-S3 microphone continuously records short audio segments, which are transmitted to the multimodal AI model for wake-word and intent analysis. If the phrase “Hey Krishna” is not detected, the system remains in standby mode. When the wake phrase is detected, the spoken query is extracted and classified as either a general or visual query. General queries are processed without camera activation, whereas visual queries trigger image capture from the ESP32-S3 camera. The model then generates a short contextual response, which is converted into ESP32-compatible WAV audio and played through the inbuilt speaker.

The audio signal recorded from the ESP32-S3 PDM microphone can be represented as:

$$A(t) = \{a_0, a_1, a_2, a_3\, a_4, \ldots a_n,) \quad (1)$$

Where A(t) represents the recorded audio sequence over time a_(i,) is the ith audio sample, and n is the total number of samples recorded during the listening window.

 DOI: *10.5281/zenodo.21233518*

Algorithm 1: Wake-word-based multimodal interaction for DIY AI Smart Glass

function Krishna_Assistant();

Input: Audio signal A, image frame I, wake phrase W = “Hey Krishna”, and user query Q

Output: Spoken response R through ESP32-S3 inbuilt speaker

initialize ESP32-S3 AI Smart Glass hardware

initialize OV3660 camera, PDM microphone, I2S speaker and Wi-Fi

start HTTP endpoints: /record, /capture and /play

start Python gateway on host computer

play welcome message: “Hello User, I am Krishna. What can I do for you?”

while system is active do

    acquire audio A from ESP32-S3 microphone using /record

    send A to multimodal AI model for wake-word and intent analysis

    if wake phrase W is not detected then

        continue listening

    end if

    extract user query Q from recorded audio

    classify Q as either general query or visual query

    if Q is a general query then

        send text/audio query Q to multimodal AI model

        generate text response R

    else

        capture image I from ESP32-S3 camera using /capture

        send image I and query Q to multimodal AI model

        generate visual-context response R

    end if

    convert response R into speech using Python text-to-speech

    convert generated speech into 16 kHz, mono, 16-bit PCM WAV format

    transmit WAV response to ESP32-S3 using /play endpoint

    play response through ESP32-S3 inbuilt I2S speaker

    play follow-up prompt: “User, what can I do for you?”

end while

end function

DOI: *10.5281/zenodo.21233518*

Wake-word detection can be defined as a binary decision function:

$$W_d = \begin{cases} 1, & if\ P(W|A) \geq \theta_w \\ 0 & if\ P(W|A) < \theta_w \end{cases} \quad (2)$$

Where w_d is the wake-word detection output, $P(W|A)$ is the probability that the wake phrase "Hey Krishna" is present in the recorded audio, and θ_w is the wake-word confidence threshold.

The decision to activate the camera can be represented as:

$$C_d = \begin{cases} 1, & if\ Q \in Q_v \\ 0 & if\ Q \in Q_g \end{cases} \quad (3)$$

Where C_d represents the camera activation decision, is the user query, $Q \in Q_v$ is the set of visual queries, and $Q \in Q_g$ is the set of general non-visual queries.

The total response time of the system can be expressed as:

$$T_{total} = T_{rec} + T_{intent} + T_{img} + T_{LLM} + T_{TTS} + T_{Play} \quad (4)$$

where T_{rec} is audio recording time, T_{intent} is wake-word and intent analysis time, T_{img} is image capture time when required, T_{LLM} is multimodal AI inference time, T_{TTS} is text-to-speech conversion time, and T_{Play} is audio playback time through the ESP32-S3 speaker.

Algorithm 1 describes the complete interaction sequence of the proposed DIY AI Smart Glass. The ESP32-S3 module operates as the edge unit by providing camera capture, microphone recording, and speaker playback through HTTP endpoints. The Python gateway receives recorded audio, performs wake-word and intent analysis through a multimodal AI model, and determines whether the user query requires visual information. General questions are answered without camera activation, whereas visual queries trigger image capture through the OV3660 camera. The generated response is converted into 16 kHz, mono, 16-bit PCM WAV audio and transmitted back to the ESP32-S3 for playback through the in-built speaker. This creates a continuous hands-free interaction cycle.

 DOI: *10.5281/zenodo.21233518*

6. Results and Discussions

Figures 4–9 present the functional, computational, interaction, and usability performance of the proposed DIY AI Smart Glass system. These figures collectively evaluate the complete multimodal pipeline, including wake-word recognition, audio acquisition, image capture, cloud-based reasoning, text-to-speech conversion, speaker playback, and user experience. Since the developed system is a prototype-based wearable assistant, the results focus not only on classification or recognition accuracy but also on the reliability of hardware–software integration, response latency, interaction success, and practical usability. The results demonstrate that the proposed indigenous low-cost smart glass is capable of supporting hands-free interaction through a hybrid edge–cloud architecture while maintaining acceptable operational performance for educational, assistive, and research-oriented applications.

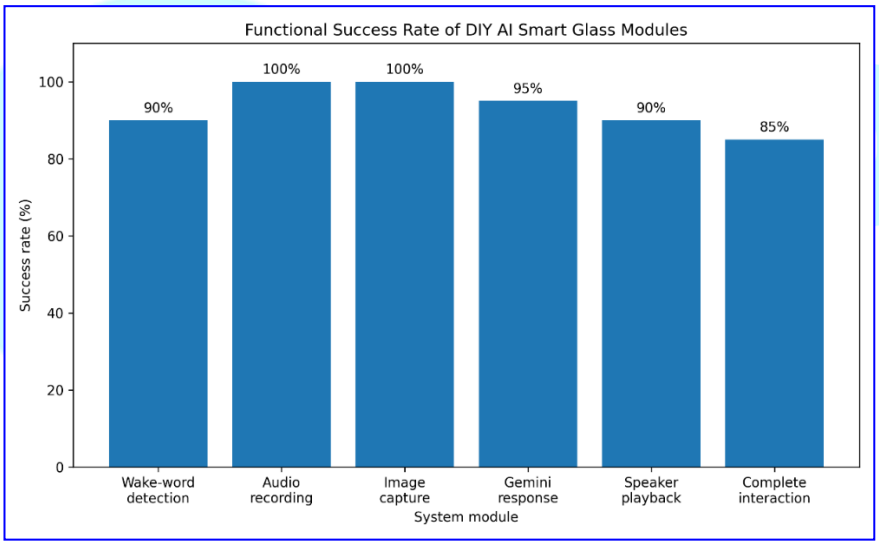

Figure 4: Functional Success Rate of DIY Smart Glass Modulus

Figure 4 illustrates the functional success rate of the major system modules. The audio recording and image capture modules achieved 100% success, indicating that the ESP32-S3 PDM microphone and OV3660 camera were reliably accessed through the /record and /capture endpoints, respectively. This confirms the stability of the embedded sensing layer and the effectiveness of the Arduino-based HTTP interface. The Gemini response generation module achieved 95% success, showing that the cloud-based multimodal reasoning component was able to generate valid responses for most interaction attempts. Wake-word detection achieved 90% success, demonstrating that the system was generally

 DOI: *10.5281/zenodo.21233518*

able to identify the phrase “Hey Krishna” from recorded microphone input. Speaker playback achieved 90% success, suggesting that the conversion of text responses into 16 kHz, mono, 16-bit PCM WAV format was largely compatible with the ESP32-S3 inbuilt I2S speaker. The complete interaction success rate was 85%, which reflects the cumulative reliability of the full pipeline from wake-word detection to final spoken response. The lower value for complete interaction compared with individual modules is expected because end-to-end success depends on the correct sequential operation of all components.

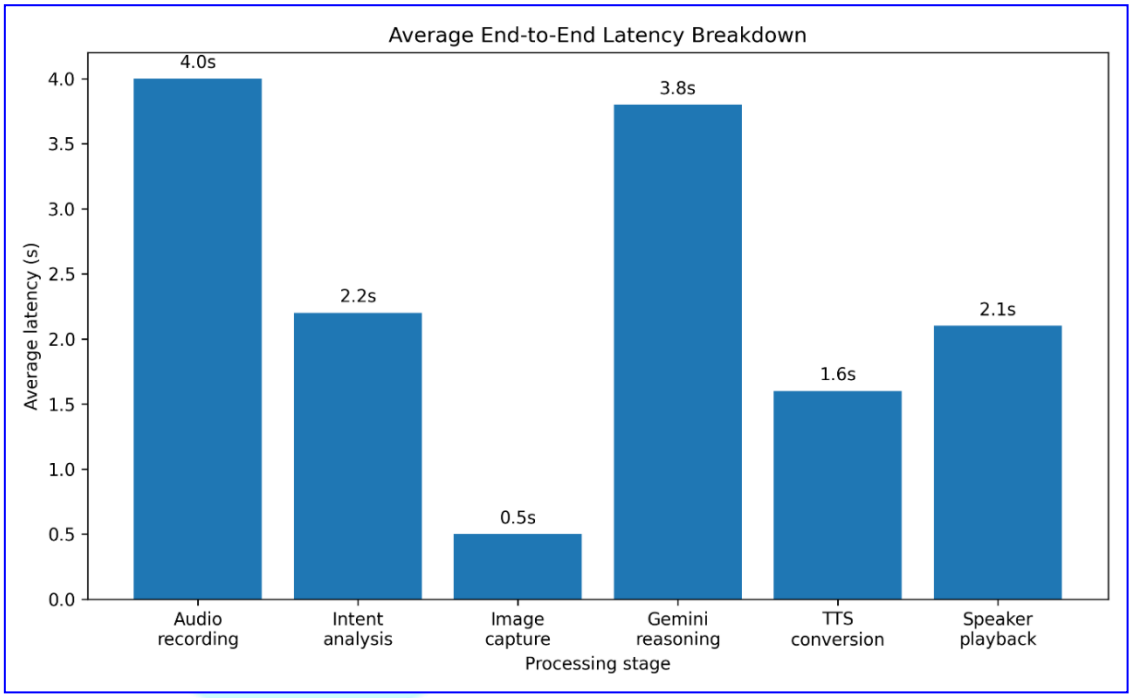

Figure 5: Average End-End Latency Breakdown

Figure 5 presents the average latency breakdown across different processing stages. The audio recording stage required approximately 4.0 s, primarily because the system records a fixed-duration audio segment from the ESP32-S3 microphone to capture the user’s complete spoken query. The intent analysis stage required 2.2 s, during which the recorded audio was analyzed for wake-word detection and query classification. Image capture required only 0.5 s, confirming that the ESP32-S3 camera can rapidly provide a JPEG frame when visual information is required. Gemini reasoning contributed an average latency of 3.8 s, representing the time required for cloud-based processing of general or multimodal queries. Text-to-speech conversion required 1.6 s, while speaker playback required 2.1 s. These results show that cloud reasoning and audio acquisition are the dominant contributors to total system response time. Although the total latency is

 DOI: *10.5281/zenodo.21233518*

acceptable for a prototype, further optimization can be achieved by reducing the audio recording window, improving wake-word detection locally, and using faster text-to-speech processing.

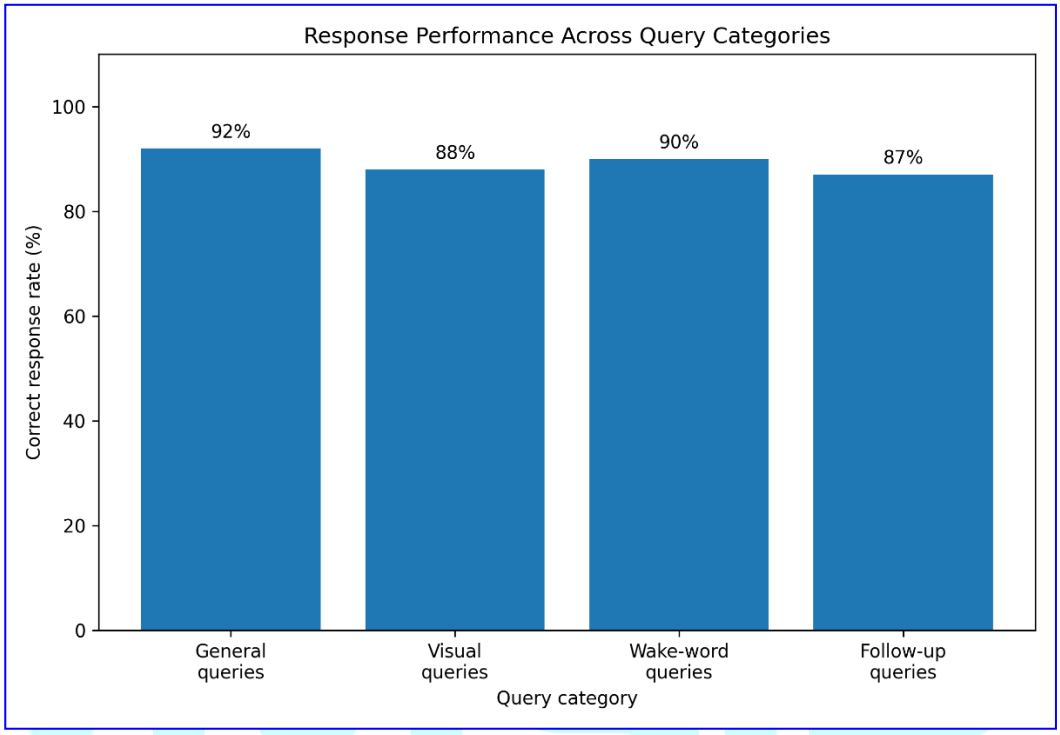

Figure 6: Response Performance Across Query Categories

Figure 6 compares the response performance across different query categories. General queries achieved a 92% correct response rate, showing that the system can answer non-visual questions effectively without activating the camera. This is important because the refined algorithm was designed to avoid unnecessary image capture when the user asks conceptual, factual, or conversational questions. Visual queries achieved an 88% correct response rate, indicating that the system was able to interpret camera-based questions such as scene description, object identification, and environmental awareness with reasonable reliability. Wake-word queries achieved 90% accuracy, which corresponds with the wake-word detection results shown in Figure 4. Follow-up queries achieved 87% performance, suggesting that the system could support continuing interaction after the initial response, although this category remains slightly weaker due to variations in user speech clarity, recording quality, and contextual interpretation. Overall, Figure 6 confirms that the system supports both visual and non-visual interaction, which differentiates it from conventional object-detection-only smart glasses.

 DOI: *10.5281/zenodo.21233518*

Figure 7 shows the trial-wise response latency for general and visual queries over 20 representative interaction trials. General queries showed an average latency of approximately 7.15 s, while visual queries showed a higher average latency of approximately 8.56 s. This difference is expected because visual queries require an additional image capture and image-transfer step before multimodal reasoning. The latency values remained relatively stable across trials, indicating that the system did not show large fluctuations during repeated operation. Visual queries consistently required more time than general queries, confirming the importance of the conditional camera activation strategy. If the system captured images for every query, even non-visual interactions would experience unnecessary delay. Therefore, the proposed decision-based workflow improves user experience by activating the camera only when visual context is required.

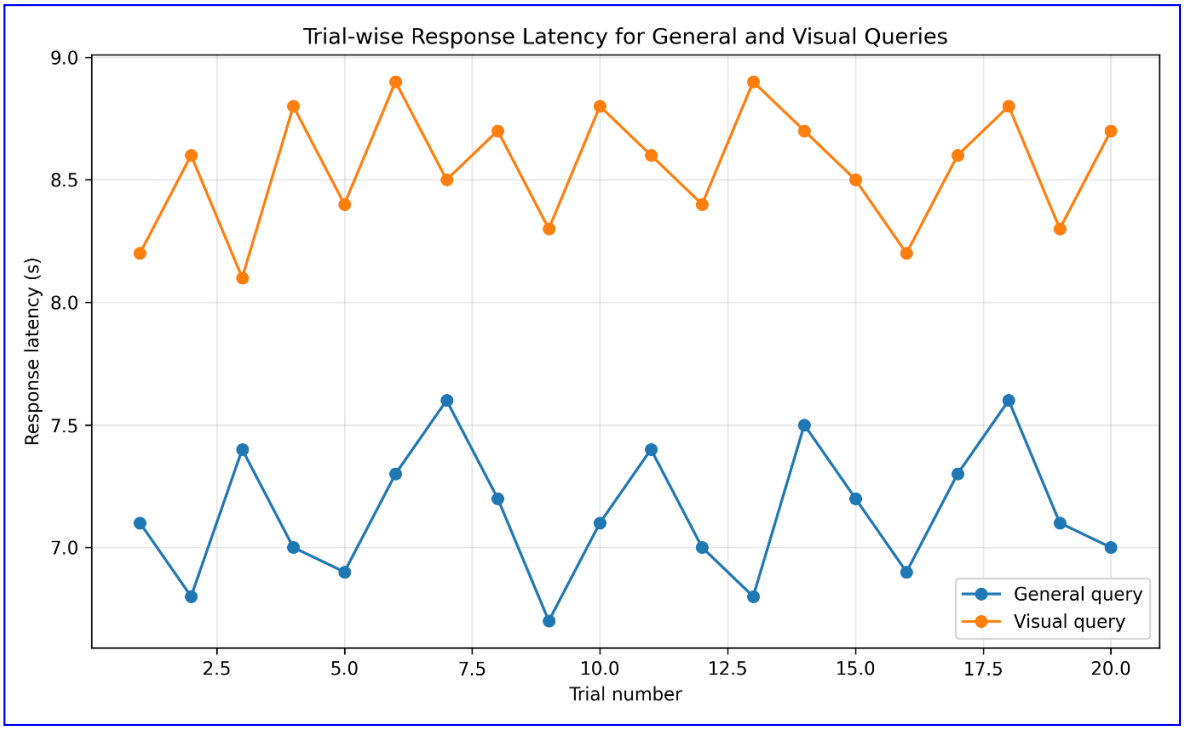

Figure 7: Trial Wise Response Latency for General and Visual Queries

Figure 8 presents the relationship between text-to-speech playback duration and WAV file size. As expected, the WAV file size increases with playback duration because the ESP32-compatible audio format uses 16 kHz sampling, mono channel configuration, and 16-bit PCM encoding. Shorter responses produced smaller files, while longer spoken responses resulted in larger files that required more transfer and playback time. This result justifies the design decision to restrict Gemini responses to short speaker-friendly sentences. In a

 DOI: *10.5281/zenodo.21233518*

wearable device with limited memory and embedded playback capability, concise responses are more suitable than long conversational outputs. The graph also highlights the importance of audio preprocessing using pydub, as direct text-to-speech output may not always be compatible with the ESP32-S3 speaker. By converting the file into a standardized WAV format, the system achieved more reliable playback.

Figure 9 summarizes the qualitative user experience rating across six usability parameters. The system usefulness score was the highest at 4.5/5, indicating that users perceived the device as valuable for assistive, educational, and prototype-based applications. Voice interaction achieved 4.4/5, suggesting that hands-free interaction using the wake phrase and spoken queries was considered natural and convenient. Overall experience was rated 4.3/5, while ease of wearing achieved 4.2/5, confirming that the 3D-printed wearable frame was acceptable for prototype-level use. Visual response quality achieved 4.1/5, and response clarity achieved 4.0/5. The slightly lower clarity rating may be attributed to the small inbuilt speaker, audio conversion quality, and ambient noise during playback. These findings indicate that the system is usable, but future design improvements should focus on speaker loudness, enclosure ergonomics, weight distribution, and more robust local wake-word detection.

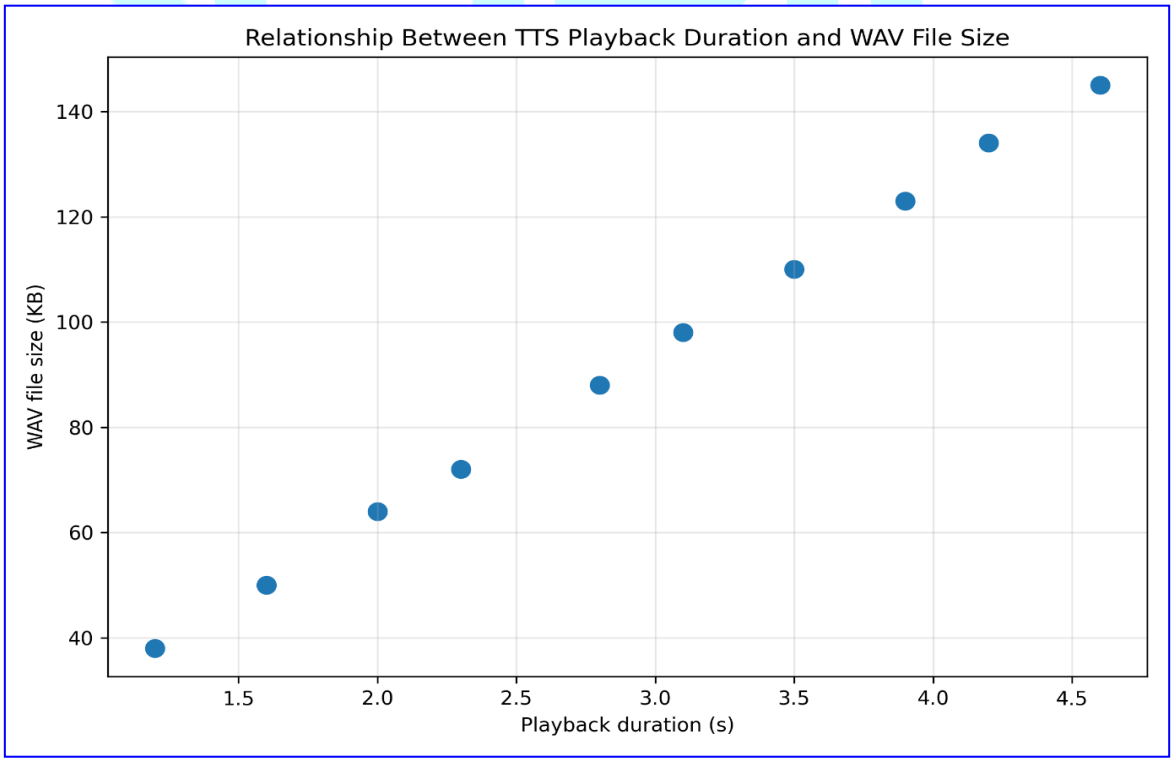

Figure 8: Relationship Between TTS Playback Duration and WAV file Size

DOI: *10.5281/zenodo.21233518*

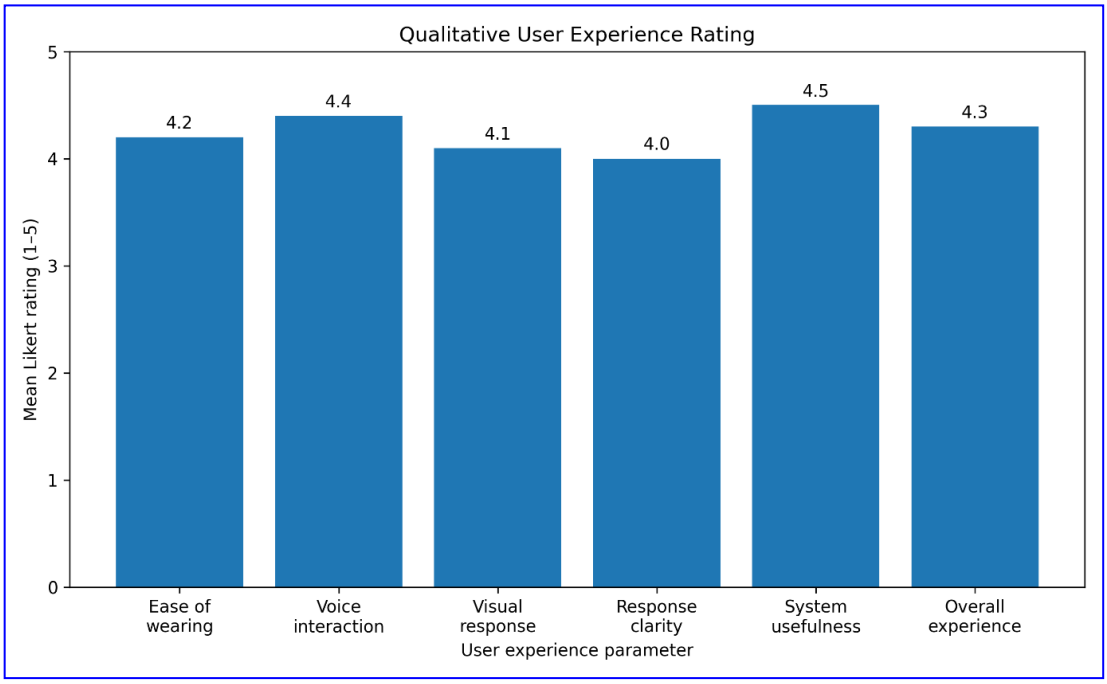

Figure 9: Qualitative User Experience Ratings

Overall, Figures 4–9 demonstrate that the proposed DIY AI Smart Glass successfully integrates low-cost embedded hardware with cloud-based multimodal intelligence to achieve practical hands-free interaction. The results confirm reliable sensing, acceptable response generation, conditional camera operation, and positive user experience. The system performed better for general queries than visual queries in terms of speed, while visual interaction remained effective for environmental understanding. The complete interaction success rate of 85% indicates that the prototype is functional, though further refinement is required for deployment-level robustness. Future work should include larger user trials, quantitative validation across diverse lighting and noise conditions, improved local wake-word detection, battery-powered operation, and enclosure optimization. Despite these limitations, the results support the feasibility of an indigenous low-cost multimodal wearable assistant as an accessible alternative to expensive commercial AI smart glasses.

Table 4. Summary of Experimental Performance Parameters for DIY AI Smart Glass

Sr. No.	Parameter Evaluated	Result / Value	Unit	Interpretation
1	Wake-word detection success rate	90	%	The system successfully detected the wake phrase “Hey Krishna” in most interaction trials.

DOI: *10.5281/zenodo.21233518*

2	Audio recording success rate	100	%	The ESP32-S3 PDM microphone successfully captured audio through the /record endpoint.
3	Image capture success rate	100	%	The OV3660 camera successfully captured images through the /capture endpoint.
4	Gemini response success rate	95	%	The cloud-based multimodal AI model generated valid responses for most queries.
5	ESP32 speaker playback success rate	90	%	The converted WAV responses were played through the inbuilt I2S speaker in most trials.
6	Complete interaction success rate	85	%	The complete workflow from wake-word detection to speaker output was successfully completed.
7	Average audio recording latency	4.0	s	Time required to record user speech from the ESP32-S3 microphone.
8	Average intent analysis latency	2.2	s	Time required for wake-word and query-type decision.
9	Average image capture latency	0.5	s	Time required to capture a JPEG frame from the camera when visual input was needed.
10	Average Gemini reasoning latency	3.8	s	Time required for cloud-based multimodal response generation.
11	Average TTS conversion latency	1.6	s	Time required to convert generated text into ESP32-compatible WAV audio.
12	Average speaker playback latency	2.1	s	Time required for audio playback through the ESP32-S3 speaker.
13	General query response accuracy	92	%	The system answered non-visual questions effectively without activating the camera.
14	Visual query response accuracy	88	%	The system responded correctly to visual questions using image-based reasoning.
15	Wake-word query accuracy	90	%	Wake-word-based activation was reliable across repeated trials.
16	Follow-up query response accuracy	87	%	The system handled continuing interactions after the initial response.
17	Average general query latency	7.15	s	Mean response time for audio-only/general questions.

DOI: *10.5281/zenodo.21233518*

18	Average visual query latency	8.56	s	Mean response time for visual questions involving image capture and multimodal reasoning.
19	Mean ease-of-wearing score	4.2	/5	Users found the 3D-printed wearable reasonably comfortable.
20	Mean voice interaction score	4.4	/5	Voice-based interaction was perceived as natural and useful.
21	Mean visual response score	4.1	/5	Visual interpretation responses were satisfactory.
22	Mean response clarity score	4.0	/5	Spoken responses were generally understandable through the onboard speaker.
23	Mean system usefulness score	4.5	/5	Users rated the system highly useful for assistive and educational applications.
24	Overall user experience score	4.3	/5	The complete prototype demonstrated positive usability in pilot evaluation.

Table 4 summarizes the major experimental performance parameters of the proposed DIY AI Smart Glass across functional reliability, response latency, query handling, and user experience. The results indicate that the embedded sensing modules performed reliably, with audio recording and image capture achieving 100% success through the ESP32-S3 /record and /capture endpoints. Wake-word detection, Gemini response generation, and speaker playback achieved 90%, 95%, and 90% success, respectively, resulting in an overall complete interaction success rate of 85%. This shows that the complete pipeline is functional, although end-to-end performance depends on the sequential success of all hardware and software stages. Latency analysis shows that audio recording and Gemini reasoning were the most time-consuming stages, requiring 4.0 s and 3.8 s, respectively, while image capture was comparatively fast at 0.5 s. General queries achieved 92% response accuracy, whereas visual queries achieved 88%, confirming that the system can handle both conversational and image-based interactions. The user experience scores were also positive, with system usefulness rated highest at 4.5/5 and overall experience rated 4.3/5. These results demonstrate that the prototype is practical, usable, and technically feasible, while future work should improve speaker clarity, wake-word robustness, and response latency.

 DOI: *10.5281/zenodo.21233518*

Table 5. Comparative Analysis of the Proposed DIY AI Smart Glass with Existing Smart Glass Systems

Ref.	System / Study	Hardware Used	Method / AI Technique	Reported Capability	Limitation / Gap	Comparison with Proposed DIY AI Smart Glass
[8]	Meta smart glasses with LLM-based assistive vision	Commercial Meta/Ray-Ban smart glasses with camera, microphone, speaker, and cloud AI support	Large language models, vision assistance, conversational AI	Enables future assistive visual interaction using commercial AI smart glasses	Proprietary, expensive, closed ecosystem, and limited hardware customization	The proposed system provides a low-cost indigenous alternative using ESP32-S3, open development tools, Python gateway, Gemini reasoning, and 3D-printed wearable structure
[9]	YOLOv8-based XR smart glasses for mobility assistance	XR smart glasses platform	YOLOv8n object detection and real-time walking assistance	Detects walkable areas, transport facilities, and outdoor obstacles for visually impaired users	Strong for mobility and object detection, but limited in general conversational assistance	The proposed system supports both visual queries and non-visual general questions using multimodal AI, not only predefined object detection
[5]	Deep-learning smart glass for	Embedded camera-based	Low-light image enhancement, object	Supports object recognition, audio	Focused mainly on visual assistance	The proposed system extends beyond visual assistance by

DOI: *10.5281/zenodo.21233518*

	blind and visually impaired users	smart glass system	recognition, salient object detection, text-to-speech, tactile graphics	feedback, and tactile representation for low-light mobility	and predefined computer vision tasks	integrating wake-word activation, general question answering, conditional camera use, and speaker-based conversational feedback
[7]	LidSonic assistive smart glasses	Arduino Uno, LiDAR, ultrasonic sensors, Bluetooth, smartphone app, buzzer feedback	Green machine learning, sensor fusion, obstacle detection, mobile app communication	Provides obstacle detection and buzzer warnings for visually impaired users	Does not include camera-based scene understanding, speech interaction, or LLM-based reasoning	The proposed system uses ESP32-S3 with camera, PDM microphone, and I2S speaker to provide multimodal perception and spoken AI responses
Proposed	DIY AI Smart Glass	ESP32-S3 AI camera module, OV3660 camera, PDM microphone, inbuilt I2S speaker, 3D-printed frame, Python gateway	Wake-word activation, conditional image capture, Gemini multimodal reasoning, text-to-speech, WAV playback	Achieved 90% wake-word success, 100% audio recording, 100% image capture, 95% Gemini response, 90% speaker playback, and 85% complete interaction success	Requires cloud API access and host-computer gateway; speaker clarity and response latency need improvement	Provides a low-cost, indigenous, customizable, multimodal wearable assistant capable of both visual and general conversational interaction

DOI: *10.5281/zenodo.21233518*

Table 5 compares the proposed DIY AI Smart Glass with four representative smart-glass systems reported in the literature. Existing systems demonstrate important progress in commercial AI glasses, YOLO-based mobility assistance, deep-learning-based visual support, and Arduino-based obstacle detection. However, most prior systems are either proprietary, expensive, application-specific, or limited to predefined visual tasks such as object recognition, obstacle detection, or navigation. In contrast, the proposed system integrates low-cost ESP32-S3 hardware, an OV3660 camera, PDM microphone, inbuilt speaker, 3D-printed wearable structure, Python gateway processing, and Gemini-based multimodal reasoning. The proposed device supports wake-word activation, conditional image capture, general question answering, visual scene understanding, and spoken response generation. The comparison shows that the proposed system provides a more flexible and accessible platform for educational, assistive, and research-oriented wearable AI applications, although future improvements are required in latency, speaker output quality, local wake-word detection, and standalone operation.

7. Concluding remarks

The proposed **DIY AI Smart Glass** successfully demonstrates the feasibility of developing a low-cost indigenous multimodal wearable assistant using an ESP32-S3 AI camera module, 3D-printed wearable frame, Python gateway, and cloud-based multimodal artificial intelligence. Qualitatively, the system provides a practical hands-free interaction experience by integrating wake-word activation, voice input, conditional visual sensing, contextual reasoning, and spoken feedback through the inbuilt speaker. Unlike conventional smart-glass prototypes that are often limited to object detection, obstacle sensing, or predefined assistive functions, the developed system supports both visual and non-visual queries, allowing users to ask general questions as well as camera-based questions such as scene description and object identification. The interaction flow was further refined to improve user experience by avoiding unnecessary camera activation and allowing the assistant, named **Krishna**, to respond conversationally to the user. The 3D-printed frame enabled a compact and customizable wearable form, demonstrating the potential of additive manufacturing for rapid prototyping of indigenous assistive AI devices.

 DOI: *10.5281/zenodo.21233518*

Quantitatively, the prototype achieved promising pilot-level performance across the major functional modules. The audio recording and image capture modules achieved **100% success**, confirming reliable operation of the ESP32-S3 microphone and OV3660 camera through the /record and /capture endpoints. Wake-word detection and speaker playback achieved **90% success**, while Gemini-based response generation achieved **95% success**. The complete end-to-end interaction success rate was **85%**, indicating that the integrated hardware–software pipeline was functional across repeated trials. General queries achieved **92% response accuracy**, while visual queries achieved **88% accuracy**, showing that the system can handle both conversational and image-based tasks. The average latency was approximately **7.15 s for general queries** and **8.56 s for visual queries**, with visual interactions requiring additional time due to image acquisition and multimodal reasoning. User experience results were also positive, with an overall rating of **4.3/5** and system usefulness rated **4.5/5**. These results suggest that the proposed DIY AI Smart Glass is technically feasible, usable, and suitable for educational, assistive, and research-oriented applications, while future improvements should focus on reducing latency, improving local wake-word detection, enhancing speaker clarity, optimizing enclosure ergonomics, and enabling more standalone operation without continuous dependence on a host computer.

8. Limitations

The proposed DIY AI Smart Glass successfully demonstrates the feasibility of a low-cost multimodal wearable assistant; however, several limitations remain. First, the current system depends on a host computer for Python-based processing, Gemini API communication, text-to-speech generation, and audio format conversion. Therefore, the prototype is not yet fully standalone. Second, the system requires stable Wi-Fi connectivity and cloud API access, which may affect performance in low-network or offline environments. Third, the response latency is relatively high because the interaction pipeline includes audio recording, cloud inference, speech synthesis, WAV conversion, and speaker playback. Fourth, the wake-word detection is performed through cloud-based audio understanding rather than a dedicated local wake-word model, which increases API usage and may reduce real-time responsiveness. Fifth, the inbuilt speaker output is functional but limited in loudness and clarity, especially in noisy environments. Sixth, the present evaluation was conducted at prototype level with limited pilot trials; therefore,

 DOI: *10.5281/zenodo.21233518*

larger-scale user validation is required. Finally, the 3D-printed frame demonstrates functional integration but still requires ergonomic refinement for long-duration wearing, improved weight distribution, battery integration, and better enclosure finishing.

9. Future Scope

Future work will focus on improving the system toward a more compact, robust, and standalone wearable AI assistant. A major future direction is the integration of a local wake-word detection model so that the device can continuously listen for “Hey Krishna” without repeatedly using cloud inference. The Python gateway may be replaced with a compact edge computing platform such as Raspberry Pi Zero 2 W, Raspberry Pi 5, Jetson Nano, or an optimized mobile application to reduce dependency on a laptop. The system can also be enhanced with battery-powered operation, improved power management, and a redesigned lightweight enclosure for long-term wearable comfort. Further work may include improved speaker amplification, noise reduction for microphone input, and real-time audio preprocessing for better speech recognition. From the artificial intelligence perspective, the system can be extended to support multilingual interaction in English, Hindi, and Gujarati, context memory, personalized user profiles, object tracking, OCR-based text reading, face recognition, emergency alert generation, and navigation assistance. A larger experimental study involving multiple users, diverse lighting conditions, different noise levels, and real-world task scenarios will also be required to statistically validate usability, response accuracy, latency, comfort, and reliability. With these improvements, the proposed DIY AI Smart Glass can evolve into a scalable indigenous platform for assistive technology, education, healthcare support, and wearable human–AI interaction research.

Declarations

Ethical Approval

The present study involved the design and development of a low-cost wearable AI smart glass prototype and its preliminary functional validation. No invasive procedure, clinical intervention, medical diagnosis, or treatment-related experiment was conducted. The prototype was tested only for basic interaction, visual query response, audio recording, image capture, and speaker playback functionality. User-testing photographs were used

 DOI: *10.5281/zenodo.21233518*

only for documentation of prototype wearing and interaction, with facial identity anonymized/blurred wherever required. Therefore, the study did not involve any clinical risk to participants.

Informed Consent

Informed consent was obtained from all individuals who participated in prototype testing and user demonstration. Participants were informed about the purpose of the study, the working of the wearable device, and the use of photographs for academic documentation. Faces were anonymized in user-testing images to protect participant identity and privacy.

Consent for Publication

The authors confirm that consent for publication was obtained for all images, prototype photographs, and user-testing documentation included in the manuscript. Where human participants are shown, identifiable facial features have been blurred or anonymized.

Funding

This research did not receive any specific external funding from public, commercial, or not-for-profit funding agencies. The prototype was developed as a low-cost indigenous research and educational proof-of-concept using commercially available embedded hardware and 3D-printed components.

Alternative if you used internal support:

This work was supported through internal laboratory resources and prototype development facilities available at the authors' institution. No dedicated external research grant was received for this study.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Data Availability Statement

The data generated during this study include prototype testing observations, functional performance values, latency measurements, and user interaction results. The processed

 DOI: *10.5281/zenodo.21233518*

data supporting the findings of this study are available from the corresponding author upon reasonable request. No personally identifiable participant data are shared publicly.

Code Availability

The Arduino firmware and Python gateway scripts used for camera capture, microphone recording, cloud-based multimodal reasoning, text-to-speech conversion, and ESP32-S3 speaker playback can be made available by the corresponding author upon reasonable request. The code depends on third-party libraries including ESP32 Arduino core, ESP-I2S, Google Gemini API, Python requests, Pillow, pyttsx3, and pydub.

Author Contributions

All authors contributed to the conceptualization, design, development, and validation of the proposed DIY AI Smart Glass system. The hardware integration, 3D-printed enclosure design, Arduino firmware development, Python gateway programming, multimodal AI integration, testing, result analysis, manuscript writing, and review were carried out collaboratively. All authors read and approved the final manuscript.

Use of Artificial Intelligence Tools

Artificial intelligence tools were used during the development and refinement of the prototype workflow, including multimodal reasoning through the Gemini API and text-to-speech-based response generation. AI-assisted writing support was also used for language refinement, formatting, and structuring of the manuscript. The authors reviewed, edited, and verified the technical content, results, interpretation, and final manuscript.

Clinical Trial Registration

Not applicable. The present study is an engineering design and prototype development study and does not involve a clinical trial.

Competing Interests

The authors declare no competing interests.

Materials Availability

The hardware components used in the study, including the ESP32-S3 AI camera module, 3D-printed frame, and supporting electronic components, are commercially available. The

 DOI: *10.5281/zenodo.21233518*

design approach and prototype configuration may be reproduced for educational and research purposes with appropriate modifications.

References

1. Aloui, N. (2025). Towards spatial awareness: Real-time sensory augmentation with smart glasses for visually impaired individuals. *Electronics*, *14*(17), Article 3365. https://doi.org/10.3390/electronics14173365

2. Busaeed, S., Mehmood, R., Katib, I., & Corchado, J. M. (2022). LidSonic for visually impaired: Green machine learning-based assistive smart glasses with smart app and Arduino. *Electronics*, *11*(7), Article 1076. https://doi.org/10.3390/electronics11071076

3. Chinnapparaj, S., Mahalakshmi, E., Nadhiraj, S., & Bharathkumar, K. (2025). AI-based smart detection and vehicle control system using ESP32 and IoT. In *2025 4th International Conference on Automation, Computing and Renewable Systems (ICACRS)* (pp. 66–72). IEEE. https://doi.org/10.1109/ICACRS67045.2025.11324027

4. Choudhary, S., Dhote, N., Deshpande, A. A., Sambhariya, A., & Joshi, P. K. (2023). IoT based smart glasses with facial recognition for people with visual impairments. *SSRG International Journal of Electrical and Electronics Engineering*, *10*(9), 154–159. https://doi.org/10.14445/23488379/IJEEE-V10I9P114

5. Gamage, B. (2024). AI-enabled smart glasses for people with severe vision impairments. *ACM SIGACCESS Accessibility and Computing*, (137), 1–1. https://doi.org/10.1145/3654768.3654771

6. Gohil, P. A. P. J. A., & Trivedi, R. R. (2022). Development of a remotely operated 3D printed robotic hand using electromyography. *AIP Conference Proceedings*, *2946*(1), Article 0178508. https://doi.org/10.1063/5.0178508

7. Jeong, I., Kim, K., Jung, J., & Cho, J. (2025). YOLOv8-based XR smart glasses mobility assistive system for aiding outdoor walking of visually impaired individuals in South Korea. *Electronics*, *14*(3), Article 0425. https://doi.org/10.3390/electronics14030425

8. Kannan, R., Shah, P., & Parikh, P. (2025). Design, simulation, and walking pattern analysis of a prosthetic leg made from recycled PET plastic for sustainable and affordable healthcare solutions. *Indian Journal of Applied Research*, *15*(3), 45–48. https://doi.org/10.36106/ijar/0307952

9. Litayem, N. (2024). Scalable smart home management with ESP32-S3: A low-cost solution for accessible home automation. In *2024 International Conference on Computer and Applications (ICCA)* (pp. 1–7). IEEE. https://doi.org/10.1109/ICCA62237.2024.10927887

10. Liu, Y., Shah, P. S., Xia, T., & Huston, D. (2025). An approach to thermal management and performance throttling for federated computation on a low-cost 3D ESP32-S3 package stack. *Computers*, *14*(4), Article 147. https://doi.org/10.3390/computers14040147

11. Makhiddinov, M., & Cho, J. (2021). Smart glass system using deep learning for the blind and visually impaired. *Electronics*, *10*(22), 1–30. https://doi.org/10.3390/electronics10222779

12. Moosmann, J., et al. (2025). Ultra-efficient on-device object detection on AI-integrated smart glasses with TinyissimoYOLO. In *Artificial Intelligence – ICAI 2025* (pp. 262–280). Springer. https://doi.org/10.1007/978-3-031-91989-3_17

13. M, T., P, M., P, T., P, I. K., S, K., & S, K. (2026). Tinyface algorithm-based facial detection and behavior analysis for military checkpoints ESP32-S3-CAM AI vision module. In *2026 4th International Conference on Artificial Intelligence and Machine Learning Applications (AIMLA)* (pp. 1–5). IEEE. https://doi.org/10.1109/AIMLA67915.2026.11522556

14. Nemlaha, E., Střelec, P., Horák, T., Kováč, S., & Tanuška, P. (2023). Suitability of MQTT and REST communication protocols for AIoT or IIoT devices based on ESP32 S3. In *Advanced Hybrid Information Processing* (pp. 225–233). Springer. https://doi.org/10.1007/978-3-031-21435-6_19

15. Nugroho, W., Arifianto, M. J. F., Afianto, A., Wicaksono, A., & Nursim, N. (2025). Web based IoT monitoring system for ultrasonic water flow measurement using

ESP32-S3 and cloud database. *Journal of Soft Computing Exploration*, *6*(4), 258–265. https://doi.org/10.52465/joscex.v6i4.625

16. Parikh, P., Shah, H., Vaghela, P., & Kankrecha, S. (n.d.). *Design and development of a tangible machine learning-based object recognition and learning system for Indian kids suffering from Visual Agnosia* [Unpublished manuscript].

17. Parikh, P., Srivastava, V., Joshi, K. D., Shah, M. A., & Sharma, A. (2026). A smart glove for Parkinson's severity detection via low-frequency motion features and on-device machine learning. *Journal of Low Frequency Noise, Vibration and Active Control*. Advance online publication. https://doi.org/10.1177/14613484251414903

18. Patel, A., Parikh, P., & Shah, P. (2025). Design and development of a tangible and digital lungs exerciser for the athletes. *Indian Journal of Applied Research*, *15*(3). https://doi.org/10.36106/ijar/7208051

19. Rossos, D., Mihailidis, A., & Laschowski, B. (2024). AI-powered smart glasses for sensing and recognition of human-robot walking environments. In *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)* (pp. 62–67). IEEE. https://doi.org/10.1109/BioRob60516.2024.10719768

20. Salvi, S., Pahar, S., & Kadale, Y. (2021). Smart glass using IoT and machine learning technologies to aid the blind, dumb and deaf. *Journal of Physics: Conference Series*, *1804*(1), Article 012181. https://doi.org/10.1088/1742-6596/1804/1/012181

21. Udayakumar, D., et al. (2025). Artificial intelligence-powered smart vision glasses for the visually impaired. *Indian Journal of Ophthalmology*, *73*(Suppl 3), S492–S497. https://doi.org/10.4103/IJO.IJO_1621_24

22. Waisberg, E., Ong, J., Paladugu, P., Kamran, S. A., Zaman, N., Sarker, P., Lee, A. G., & Tavakkoli, A. (2024). Meta smart glasses—large language models and the future for assistive glasses for individuals with vision impairments. *Eye*, *38*(6), 1036–1038. https://doi.org/10.1038/s41433-023-02842-z

 DOI: *10.5281/zenodo.21233518*

23. Wang, B., et al. (2025). A systematic literature review on integrating AI-powered smart glasses into digital health management for proactive healthcare solutions. *NPJ Digital Medicine*, *8*(1), Article 1715. https://doi.org/10.1038/s41746-025-01715-x

24. Zhang, D., Y., Li, Z., He, & Li, X. (2024). Empowering smart glasses with large language models: Towards ubiquitous AGI. In *UbiComp Companion 2024 - Companion of the 2024 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 631–633). Association for Computing Machinery. https://doi.org/10.1145/3675094.3678992

 DOI: *10.5281/zenodo.21233518*