# An Edge AI–Enabled Assistive Device for the Visually Impaired

Chaitanya Mule, Dweep Vartak, Devang Wangde, Prof. Vidya Sagvekar
Department of Artificial Intelligence and Data Science
K. J. Somaiya Institute of Technology, Mumbai, India
chaitanya.mule@somaiya.edu, dweep.vartak@somaiya.edu, devang.w@somaiya.edu, vsagvekar@somaiya.edu

***Abstract*—This paper presents a compact, Edge AI-powered wearable assistive device tailored to enhance the daily lives of visually impaired individuals. Leveraging the Raspberry Pi 5 in combination with the Hailo-8L AI accelerator, the system delivers responsive assistance through features such as navigation guidance, visual question answering (VQA), and general knowledge retrieval. A high-resolution camera captures live scenes, which are processed entirely on-device using optimized deep learning models to ensure low latency and privacy. The interpreted information is then converted into natural-sounding audio feedback, delivered via a speaker or Bluetooth headset, enabling users to interact more effectively with their surroundings. Designed for robustness in diverse real-world environments, this system offers a cost-effective, offline, and energy-efficient solution with significant potential to enhance mobility, awareness, and independence for the visually impaired.**

***Index Terms*—Edge AI, wearable assistive technology, Raspberry Pi 5, Hailo-8L, visual question answering, navigation aid, on-device inference, accessibility, deep learning.**

## I. Introduction

The ability to perceive and interpret one's environment is essential for independent mobility and daily life. However, for individuals with visual impairments, performing tasks such as navigating unfamiliar spaces, accessing printed or digital content, and recognizing people or objects presents persistent challenges. According to the World Health Organization (WHO), over 285 million people worldwide experience some form of visual impairment, including 39 million who are completely blind. These limitations significantly affect autonomy, often increasing dependence on caregivers and diminishing overall quality of life.

While traditional mobility aids like white canes and guide dogs offer partial assistance, they lack the capacity for interactive, context-aware information delivery. This gap has driven the growing demand for intelligent assistive technologies that can provide real-time, adaptive support in a variety of environments.

In response to this need, we propose an Edge AI-based wearable assistive device specifically designed to enhance the independence of visually impaired individuals. Unlike cloud-dependent systems that may suffer from high latency, connectivity issues, or privacy concerns, our approach utilizes the Raspberry Pi 5 (8GB) in combination with the Hailo-8L AI accelerator to perform on-device inference efficiently. This architecture guarantees low-latency operation, strong offline performance, and better data privacy—all important for carrying out real-world tasks on a wearable device.

The wearable device constantly utilizes a high-resolution camera to obtain visual input, which is processed locally using deep learning-based computer vision and natural language processing (NLP) models. These capabilities include navigation assistance, visual question answering (VQA), text reading, and retrieving general knowledge, relayed through a speaker or Bluetooth headset in intelligible audio output.

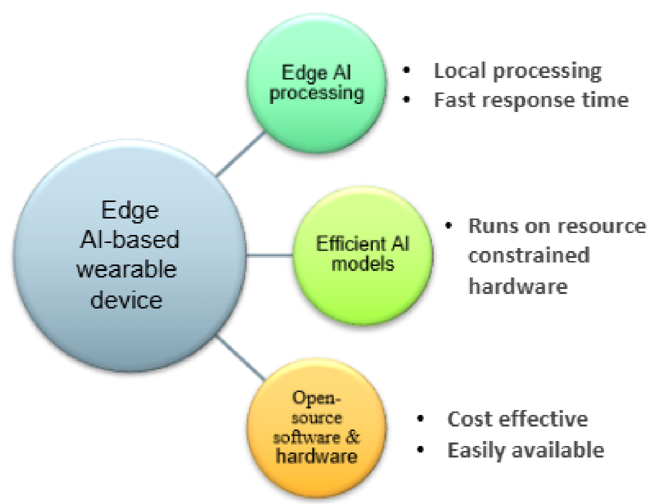


Fig. 1: Key Characteristics of an Edge AI-based Wearable Assistive Device. The system mainly focuses on local processing, efficient AI models, and cost-effective open-source hardware and software.

An important advantage of the proposed system is its robustness in a variety of real-world settings, such as indoors or outdoors, in different light levels, and with little to no connection. Its lightweight models and open-source systems support customizable and scalable deployments of the system with minimal energy cost and consumption.

A combination of rapid perception, responsive audio interaction, and a compact wearable form factor, this device could improve accessibility to individuals who are blind or have low vision. It ultimately allows for more independent navigation and information access, which increases inclusion, mobility, and quality of life.

## II. Literature Review

Assistive technology for visually impaired individuals has experienced significant advancements over recent years, primarily driven by the integration of wearable devices, deep learning models, and sensor-based navigation systems. Early efforts in this domain focused on developing sensor-based navigation aids to improve independent mobility. For instance, Munteanu and Ionel [3] presented a voice-controlled smart assistive device that utilizes ultrasonic echolocation combined with haptic and audio feedback. Although effective for short-range obstacle detection, their approach was limited by the inherent constraints of ultrasonic sensors, particularly in complex and textured environments.

Building on these foundational ideas, subsequent research explored the incorporation of computer vision and deep learning to enhance environmental perception. Kulkarni et al. [2] proposed an intelligent assistive navigator that incorporated image processing methods with deep learning systems to run on a Raspberry Pi 1. In their work, they combined various sensors, such as ultrasonic sensors to detect obstacles and water sensors to detect hazardous surfaces, to notify participants in real-time when faced with obstacles and dangerous surfaces. The intrinsic issues of sensor fusion and scalability in real-world applications continued to be associated with the system.

Simultaneously, other works in the literature have focused on using advanced vision systems to improve scene understanding. Lin et al. [5] proposed a deep learning-based wearable assistive system that used an RGB-D camera to help produce semantic maps to communicate a three-dimensional representation of the surroundings to users who are blind or visually impaired. The device did not only facilitate obstacle detection but it also allowed for more interactive assistance through a smartphone app that enabled communication between the users and assistant.

The literature also includes systems that incorporate several modalities to improve assistance. As well, some additional work, Khan et al. [6], proposed an AI-based visual aid that included an integrated reading assistant so that the system utilized a small computing unit based on a Raspberry Pi 3, simultaneously supporting users with obstacle avoidance, reading. Khan et al.'s study demonstrates the benefits of pairing a vision-based algorithm with traditional sensor methods so the overall user experience is enhanced.

More recently, the VEye-AI Vision Assistant developed by Joy et al. [1] and the wearable device created by J. Ai et al. [7] have introduced the next generation of assistive technology by utilizing advanced artificial intelligence algorithms. VEye-AI combines an AI-powered camera with an earpiece to provide audio descriptions of the surrounding environment in real-time and has additional functions including face recognition, GPS, and emergency alerts. Likewise, J. Ai et al. [7] developed a wearable device that uses image captioning techniques to perform image-to-text transformation that allows blind users to obtain contextual information about their real world environment and images on screens.

Overall, these studies show a clear progression form primitive sensor-driven technology to advanced AI-based wearables with effective environmental awareness. While this is significant progress, challenges such as sensor fusion, reducing inference latency, and ensuring user-friendly functionality all remain important future directions in research. Continued access to low-power AI hardware combined with deep learning models will potentially allow for greater attention towards the continued pursuit of a balance between accessibility and autonomy among visually impaired users.

## III. System Overview

The proposed system is a wearable AI-powered vision assistant designed to assist visually impaired users in understanding their surroundings. It leverages deep neural network models optimized for efficient and fast performance on edge devices to provide assistance. The device captures visual data, processes it locally on an edge AI platform, and provides meaningful feedback through speech output. This system aims to enhance the independence of visually impaired individuals by offering low-latency, context-aware information about their environment.

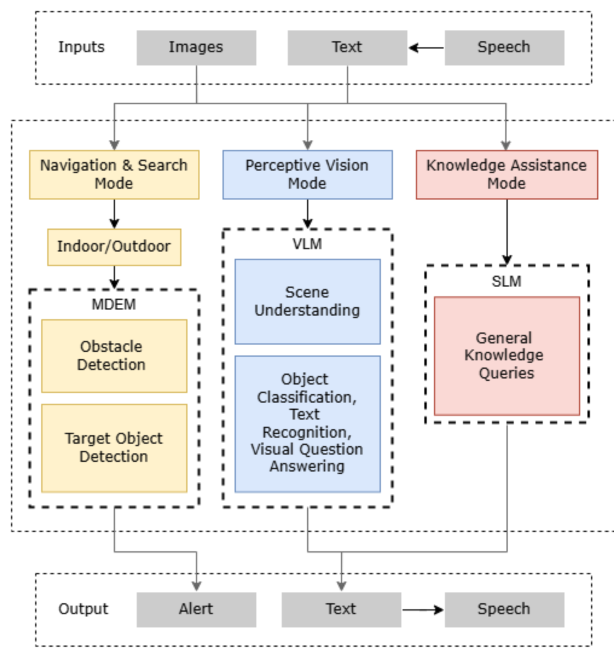


Fig. 2: Functional Overview of the Wearable Assistive System.

### A. High-Level System Architecture

The system architecture is divided into three primary components: the **hardware layer**, the **software layer**, and the **interaction layer**. These components work together to process and interpret visual information, then deliver the results to the user via audio feedback.

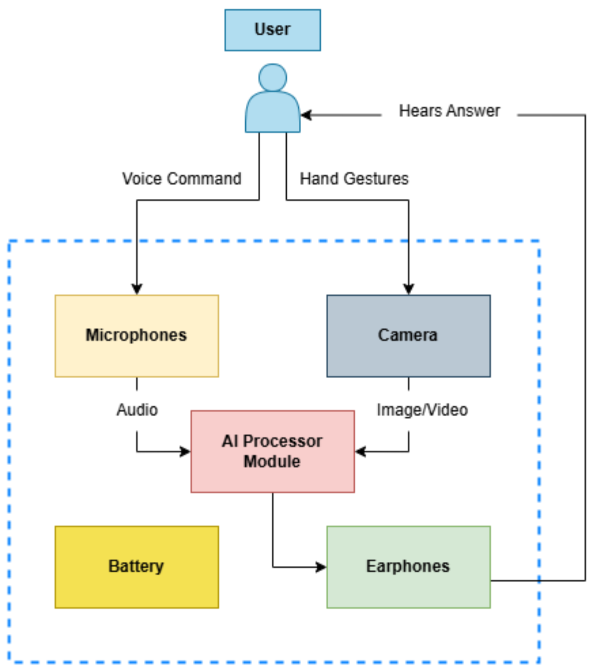


Fig. 3: High-Level Architecture of a Wearable Assistive System

*1) Hardware Layer:* The hardware layer consists of the following key components, as illustrated in **Figure 3**:

- **Camera Module:** A 12MP camera module captures real-time images of the user's surroundings.
- **AI Processor Module (Raspberry Pi 5 with Hailo-8L):** The processing unit comprises a Raspberry Pi 5 paired with the Hailo-8L AI accelerator. The Raspberry Pi handles high-level system tasks and manages communication between components, while the Hailo-8L provides specialized AI hardware for efficient computer vision tasks.
- **Audio Input and Output Modules:** The system includes microphone to capture voice commands from the user. These audio signals are processed by the AI Processor Module to interpret user intent. For output, earphones (either wired or Bluetooth-enabled) are used to provide audio feedback, delivering synthesized responses to the user.
- **Battery:** A rechargeable high capacity Li-ion battery powers the system, ensuring long operational hours while keeping the device portable

*2) Software Layer:* The software stack consists of several key modules that enable the system to interpret user queries and the captured visual data. Each module is designed to work seamlessly to deliver accurate and context-aware assistance to the user:

- **Vision Language Model (VLM):** This model employs methodologies from computer vision and natural language processing to perceive visual scenes relative to user queries. It enables the system to recognize and describe objects, text, and people in the surroundings, facilitating the user's comprehension of their environment.
- **Depth Estimation Model (DEM):** This model utilizes depth detection methodologies to estimate the distance and arrangement of objects in the user environment. It provides depth data that assist the system in navigation tasks and improving object detection capabilities including obstacle avoidance or understanding a scene.
- **Small Language Model (SLM):** The Small Language Model is a small footprint NLP model intended for use on edge devices; it can intelligently interpret user commands and generate text for general question answering. It allows the system to respond to all manner of inquiry, including fact-based ("What is the capital of France?") questions. The model is designed to balance responsiveness with accuracy, ensuring fast and accurate responses during real-time interactions with assistive technology.
- **Natural Language Processing (NLP):** A custom NLP model accurately interprets user queries and extracts intent from written or spoken text to generate actionable tasks. The model enables the system to understand the user and to respond meaningfully, providing relevant and context-appropriate feedback.
- **Speech-to-Text (STT):** The speech-to-text module enables the system to listen to user commands and convert them into text, allowing the system to process voice-based input and interpret the user's instructions/queries. This module is essential for hands-free interactions and allows the system to handle complex types of commands.
- **Speech Synthesis:** The text-to-speech (TTS) models use the generated responses and convert them to spoken words, enabling the system to provide feedback to the user in an easy-to-consume format. The TTS systems can be access through various systems like Google TTS or offline models, like Piper TTS, based on a decision of local or cloud processing.
- **AI Inference (Onnx with Pi 5):** The system uses Onnx as an environment to deploy optimized deep learning frameworks on Raspberry Pi 5 hardware. The Hailo-8L AI accelerator performs local processing of compatible models, providing an efficient inference engine with lower latency. With this architecture, we deliver edge-computed high-performance applications that allow the system to respond quickly to user queries and requests.

*3) Interaction Layer:* The interaction layer facilitates user communication with the system:

- **User Commands:** Users interact with the system through voice commands or physical gestures, requesting specific actions (e.g., "Read this text," "What's in front of me?").
- **Feedback:** The system processes the user's commands, interpreting visual data captured by the camera through AI models, and provides immediate, relevant feedback. This feedback can include the identification of objects, reading of text, or descriptions of the surrounding environment. The feedback is then converted into speech via

the speech synthesis module, delivering information in an accessible audio format.

### *B. System Workflow*

The proposed wearable device enables visually impaired users to seamlessly interact with their surroundings. As shown in **Figure 4**, the workflow follows a structured sequence to ensure a smooth user experience, from input to output.

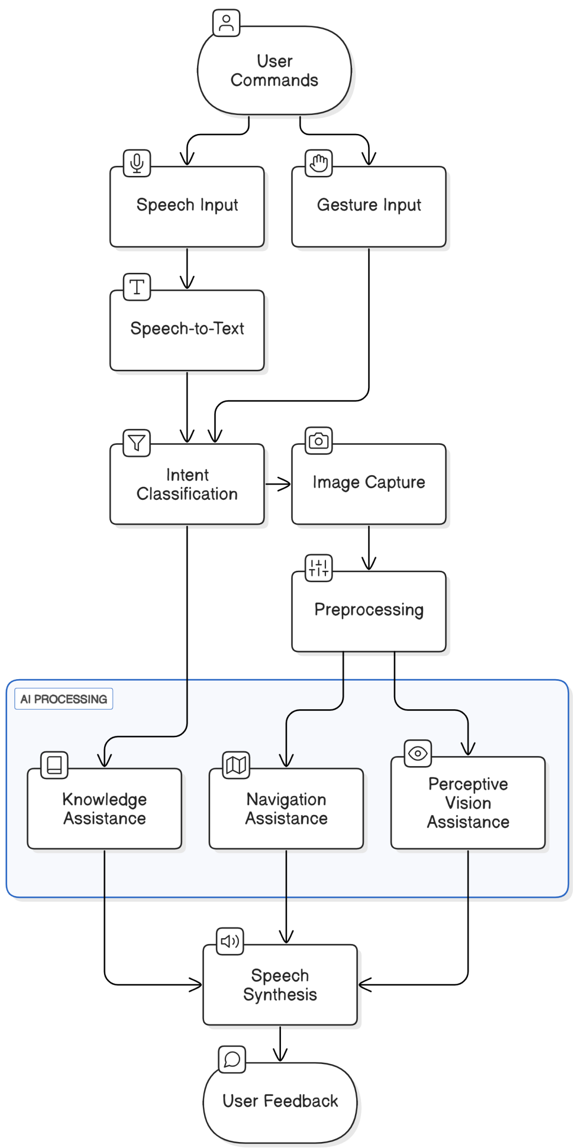


Fig. 4: Workflow of the Wearable Assistive System

- **User Initiates Interaction:** The user provides input through *voice commands or gestures* to request assistance
- **System Captures & Interprets Input:**
  - If the input is *speech-based*, it is converted to text and classified into **Navigation, Perceptive Vision, or Knowledge Assistance**.
  - If *visual input is required*, the camera captures an image for processing.
- **AI Model Processing & Decision-Making:** Based on the user's intent:
  - Navigation assistance analyzes depth and obstacles for directional guidance.
  - Perceptive vision recognizes objects, text, or requested items.
  - Knowledge assistance retrieves relevant information from a lightweight AI model.
- **Feedback Generation:** The system synthesizes a response and delivers it via **audio output (speech synthesis)** or **alert tones** for responsive assistance.

This structured workflow ensures **fast, hands-free operation**, optimizing AI processing on an edge device for **low latency and high accuracy**.

### *C. Data Flow*

The **Data Flow** represents the technical flow of data between system modules, ensuring efficient processing. Key data interactions include:

*1) Input Processing & Intent Recognition:*

- **User Input** → *Speech-to-Text Module* (Converts spoken input into text)
- **Text** → *Intent Classification Module* (Determines whether the request is for **Navigation**, **Perceptive Vision**, or **Knowledge Assistance**)

*2) Image Processing & AI Inference:*

- **If visual input is needed:**
  - **Camera Capture** → *Preprocessing Module* (Enhances image clarity and reduces noise)
  - **Preprocessed Image** → *AI Models for Inference*:
    - ∗ **Object Detection / Text Recognition** → Vision-Language Model or OCR Module (for scene understanding or reading text)
    - ∗ **Depth Estimation** → Depth Model (for navigation-related tasks such as obstacle avoidance)

*3) Decision-Making & Response Generation:*

- **AI Inference Results + User Intent** → *Decision Engine* (Maps processed data to appropriate actions—navigation guidance, perceptive vision assistance, or knowledge retrieval)
- **Decision Output** → *Response Generation Module* (Formats a meaningful, user-friendly response)

*4) Output Delivery & User Feedback:*

- **Generated Response** → *Speech Synthesis Module* (Generates natural speech from text)
- **Speech Output** → *Audio Device* (Delivers the message through a speaker or Bluetooth earpiece)

This structured data flow ensures that all components operate efficiently, leveraging **edge AI inference for quick decision-making** while minimizing latency and processing delays.

## IV. Tools and Technologies Utilized

The core system was developed on the **Raspberry Pi 5 (8GB RAM)**, enhanced by the **Hailo-8L AI module** for real-time, power-efficient AI processing at the edge. The hardware and software selections were optimized to meet the constraints the assistive devices, emphasizing low power consumption, real-time performance, and compactness.

### *1. Hardware and Software Stack*

- **Hardware:**
  - Raspberry Pi 5 (8GB RAM) serving as the main computing platform.
  - Hailo-8L AI Accelerator – dedicated to running deep learning inference efficiently.
  - 12MP Camera Module – utilized for the capture and analysis of scenes in real time.
- **Operating System:** Raspberry Pi OS (64-bit Lite) – chosen for its lightweight architecture, low boot time, and reliability in edge applications.
- **Programming Language:** Python 3.11 – selected for its extensive library support for AI/ML use cases, and hardware interaction is easier for the developer.

### *2. AI Model Formats*

- **ONNX (Open Neural Network Exchange):**executed for INFERENCE, depth 175, on Raspberry Pi with ONNX Runtime.
- **HEF (Hailo Executable Format):** an optimal format for a model executed on the Hailo-8L with HailoRT run time.

### *3. Speech Processing*

- **Speech-to-Text (STT):** Faster Whisper (ONNX) – chosen for accurate transcription of voice input, even in noisy environments.
- **Text-to-Speech (TTS):** Piper TTS (ONNX) – used for generating natural-sounding speech responses while operating offline.

### *4. Computer Vision and Inference*

- **OpenCV:** Handles camera interfacing, image preprocessing, and region-of-interest (ROI) extraction.
- **ONNX Runtime:** Executes ONNX models on the Raspberry Pi CPU for AI tasks not supported by Hailo-8L.
- **HailoRT:** The official runtime used for executing HEF models on the Hailo-8L. Python integration was performed via `hailo_platform.py`.

### *5. File and Memory Management*

- **SQLite:** Stores user preferences, logs, and session metadata in a lightweight database format.
- **Threading & Multiprocessing:** Enables efficient parallel execution of tasks such as capturing input, running inference, and generating output, ensuring responsive and smooth system operation.

## V. Model Integration and Deployment

This system exclusively used **pretrained models** to streamline development and ensure optimized real-time deployment on edge hardware.

### *1. Pretrained Model Selection*

Models were sourced in formats compatible with both CPU and AI accelerator deployment:

- **ONNX:** Models executed on the Raspberry Pi CPU using ONNX Runtime.
- **HEF:** Hailo-optimized models deployed on the Hailo-8L accelerator.

The following models were used:

- **Vision-Language Model (VLM):** `HuggingFaceTB/SmolVLM-256M-Instruct (ONNX)` for multimodal scene understanding.
- **Large Language Model (LLM):** `HuggingFaceTB/SmolLM-360-Instruct (ONNX)` for natural language processing and response generation.
- **Monocular Depth Estimation Model (MDEM):** `The scdepthv3 (HEF)` model is employed for spatial perception using input from a single RGB camera.
- **Object Detection:** YOLOv5 models from the Hailo Model Zoo (HEF) for detecting environmental objects.
- **Speech Models:**
  - **Faster Whisper (ONNX):** Converts voice commands into text.
  - **Piper TTS (ONNX):** Converts textual responses into speech.

### *2. Deployment on Hailo-8L (HEF Models)*

- Models were downloaded in `.hef` format from the Hailo Model Zoo.
- Execution was handled using the HailoRT runtime, with Python-based integration.
- Inference was offloaded to the Hailo-8L for low-latency AI processing while reducing Raspberry Pi CPU load.

### *3. Deployment on Raspberry Pi CPU (ONNX Models)*

- ONNX Runtime was used for executing models not supported by the Hailo-8L.
- This method ensured fallback inference capabilities during development and testing.

### *4. Hybrid Execution Strategy*

- The system dynamically allocated inference tasks to either the Hailo-8L or the Raspberry Pi CPU based on model compatibility and system workload.
- This hybrid approach optimized performance while maintaining efficient resource utilization.

## VI. Optimization Techniques

To ensure responsive operation on resource-constrained hardware, multiple optimization strategies were employed.

*1. Multithreading and Multiprocessing*

Python's concurrency features enabled parallel execution of various system components, significantly improving efficiency and reducing inference latency.

- **Thread 1:** Captures voice input and runs the STT engine.
- **Thread 2:** Captures and preprocesses image frames from the camera.
- **Thread 3:** Performs AI inference (VLM, LLM, depth estimation) on the appropriate hardware (Hailo-8L or CPU).
- **Main Thread:** Processes responses, generates output, and executes TTS synthesis.

This non-blocking execution flow ensures smooth interaction and minimizes response delays, making the system well-suited for wearable assistive applications.

*2. Model Quantization and Optimization*

- **Hailo-Optimized Models:** Models running on the Hailo-8L were pre-quantized for low-power, high-speed inference.
- **ONNX Model Optimization:** On the Raspberry Pi, we applied techniques like weight pruning and optimised the ONNX graph to minimise computational overhead.

*3. Efficient Resource Management*

- **Memory Constraints:** The model was optimised for execution and storage in a method that fit the constraints of the Raspberry Pi's 8GB of system RAM.
- **Adaptive Power Management:** The system was capable of adjusting the processing load on a hybrid CPU/GPU dynamically, so that we could balance power efficiency and system performance.

Altogether, these optimizations enable responsive, power-efficient performance, ensuring the system remains practical and effective for continuous, on-the-go assistive use.

## VII. Results & Discussion

The developed wearable assistive system was successfully implemented using a Raspberry Pi 5 integrated with a 12MP camera module, an audio output module, and a Hailo-8L AI processor to enhance on-device inference performance. The system is designed to assist visually impaired users with various vision-related tasks.

As shown in **Figure 5**, the prototype is compact and optimized for wearable usage. The camera module is mounted on a pair of glasses, while the processing unit—comprising the Raspberry Pi and Hailo-8L—is enclosed in a lightweight case that can be conveniently carried in a pocket or clipped to a belt. This ergonomic and modular design ensures hands-free operation, user comfort, and ease of use, making the device accessible and practical for daily activities.

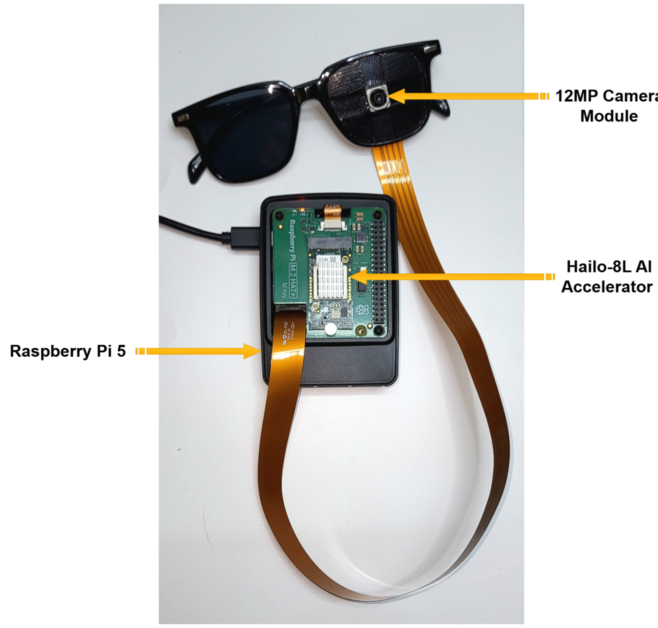


Fig. 5: Developed Prototype of the Wearable Assistive System

**Navigation & Search Mode**: As shown in **Figure 2**, the user initiates a voice command to locate a target object—for example, "Find the door." The system continuously processes the camera feed as the user scans their surroundings. When the specified object is detected with confidence above a predefined threshold, the system delivers audio feedback to indicate its presence and orientation, helping guide the user toward it. The user is informed accordingly if the object is not found.

This mode also features an obstacle detection mechanism built using a Monocular Depth Estimation Model (MDEM) that estimates distances to nearby obstacles and alerts the user when they are carefully proximate to the objects. This improves safety during navigation, especially in cluttered regions.

**Perceptive Vision Mode**: In this mode, the system utilizes a Vision-Language Model (VLM) to provide context about the scene. For example, the system may voice output information such as: "There is a table with a laptop, a mouse, and a water bottle." This provides potential for users to learn about their environment through auditory information. They can also comfortably request open-ended visual questions such as "Is there a chair nearby?" or "What is this object?" The system will provide an answer for the user based on the model that interprets the visual input.

In addition to identifying objects and contexts about scenes, the VLM can read visible environmental text such as signs, product labels, and book covers (when the text is visible and well captured). It does not read as well as a task-specific OCR module, but it will read well enough for everyday tasks that require understanding text in the environment—effectively combining the tasks of understanding vision and language. The benefit of this approach is the reduced burden on the architecture of the device without impacting achieving the functionality of reading.

**Knowledge Assistant**: A voice-based assistant utilizing a Small Language Model (SLM) gives users the ability to ask general knowledge questions and receive an answer in speech form. Knowledge Assistant has the advantage of accessibility because it quickly allows access to knowledge without visual interaction.

This multi-modal system allows users to better perceive their environment and navigate with confidence in both static and dynamic environments. Designed with a focus on accessibility and simplicity, the interface is intuitive and requires no specialized training.

*User Study & Evaluation*

To assess the usability and effectiveness of the system, a small-scale user study was conducted simulating visual impairment. Ten sighted participants were blindfolded and asked to perform a series of common assistive tasks using the prototype. Key metrics such as system response time (from command to output), total user interaction time, and subjective feedback were recorded.

TABLE I: User Study Results Summary

| **Task** | **Success Rate** | **System Response Time (s)** | **User Time (s)** | **User Feedback** |
|---|---|---|---|---|
| Find the Door | 80% | 1.8 | 2.5 | Accurate & helpful |
| Avoid Obstacles | 90% | 0.5 | 1.0 | Timely and reliable alerts |
| Scene Description | 70% | 5.6 | 12.0 | Informative and contextual |
| Ask Visual Questions | 60% | 5.3 | 9.0 | Easy to interact with |
| General Questions | 80% | 2.2 | 8.0 | Convenient and engaging |

Participants reported that the system was easy to use and provided valuable support in navigation and information gathering tasks. Some feedback suggested minor improvements in voice clarity and speaker volume under noisy conditions.

The findings confirm the system's practical applicability and its effectiveness in supporting users with visual impairments. Even in its prototype form, the device demonstrates a promising blend of responsiveness, functionality, and user-friendly design, suitable for real-world assistive applications.

## VIII. Challenges and Limitations

Despite the successful development and integration of the wearable assistive system, several challenges and limitations were encountered during the design, deployment, and testing phases. These affected system performance, user experience, and real-world applicability.

### *A. Hardware Limitations*

While the Raspberry Pi 5 provides a compact and capable edge computing platform, it struggled to concurrently execute multiple computationally demanding AI modules such as Vision-Language Models (VLMs), Language Models (LMs), Text-to-Speech (TTS), Speech-to-Text (STT), and Monocular Depth Estimation Models (MDEMs). These modules require efficient multi-threaded handling of camera input, model inference, and audio output.

To address this, the Hailo-8L AI accelerator was used to offload vision tasks, significantly enhancing inference speed and responsiveness. Models were further optimized and converted to ONNX format to reduce processing overhead. However, constraints in memory bandwidth and thermal dissipation on the Raspberry Pi 5 continued to limit sustained multitasking, especially during prolonged usage.

### *B. Model Optimization and Compatibility*

Adapting pre-trained AI models for the Hailo-8L posed several challenges. The conversion process demanded structural modifications to conform to the Hailo compiler's constraints, often requiring retraining or substitution with simpler architectures. This restricted the use of state-of-the-art models and introduced compromises in accuracy or robustness.

### *C. Latency and System Synchronization*

Ensuring low-latency performance across various subsystems—visual recognition, voice processing, and feedback delivery—proved complex. Synchronizing real-time inputs from the camera and microphone with parallel model inferences required careful thread management and system-level optimization to prevent delays and resource contention.

### *D. Environmental Sensitivity*

The performance of Perceptive Vision and Navigation Modes was significantly influenced by dynamic environmental conditions. Poor lighting, glare, motion blur, and visually cluttered backgrounds often reduced object detection accuracy and scene understanding, particularly in outdoor or fast-moving contexts.

### *E. Audio Clarity in Noisy Conditions*

Voice command recognition and audio output were frequently affected by ambient noise. Although basic noise suppression techniques were used, environments with high background noise—such as traffic or crowds—still posed significant challenges. Future work could incorporate beamforming microphones, advanced VAD (Voice Activity Detection), or deep learning-based noise cancellation.

### *F. Thermal and Power Constraints*

Continuous processing of vision and audio tasks resulted in elevated power consumption and thermal buildup. This impacted battery life, device longevity, and user comfort—critical aspects for wearable systems. In some cases, thermal throttling reduced processing speed, highlighting the need for efficient thermal management solutions such as passive heat sinks or compact active cooling systems.

### G. User Calibration and Personalization

The need for user-specific adjustments—including camera alignment, feedback volume, and response latency—presented challenges in creating a universally adaptable device. Providing simple, intuitive tools for on-device calibration remains essential for enhancing accessibility and usability across diverse users.

*Summary of Limitations*

These challenges highlight current limitations of the system in real-world usage. Environmental variability, hardware constraints, and limited model compatibility affect system robustness and scalability. Additionally, audio clarity in noisy environments, thermal performance, and lack of adaptive personalization limit user experience. Addressing these areas in future iterations will be key to improving reliability, efficiency, and inclusiveness of wearable AI-based assistive devices.

## IX. Conclusion

This paper presented the design and implementation of an edge AI-based wearable assistive device aimed at enhancing the autonomy and quality of life for individuals with visual impairments. Leveraging the computational power of the Raspberry Pi 5 in combination with the Hailo-8L AI accelerator, the system performs on-device inference for essential assistive tasks, including navigation assistance, perceptive vision, and contextual knowledge delivery. By deploying optimized deep learning models, the system delivers responsive perception and interaction without dependence on cloud services.

The implementation underscores the advantages of edge computing in wearable assistive technologies, particularly in terms of reduced latency, enhanced data privacy, and consistent functionality in network-limited environments. Through the use of optimized AI models, the device effectively addresses the inherent computational and power constraints of embedded hardware platforms.

Despite these achievements, several challenges remain, including thermal regulation, efficient multi-modal task management, and robustness under varying environmental conditions. These areas offer opportunities for further refinement and innovation.

Future development will prioritize improvements in energy efficiency, the integration of multi-camera systems, the adoption of more powerful processing hardware, and the incorporation of advanced AI models for greater contextual understanding. Additionally, extensive user studies will be conducted to assess usability, personalize system responses, and ensure alignment with diverse user needs.

By continuously evolving the capabilities of AI-driven wearable solutions, this research contributes to building a more inclusive, accessible, and intelligent ecosystem for visually impaired individuals.

## References


[1] A. Joy, N. M. Gregory, N. A. Nazeer, and J. Davis, ”VEye-AI Vision Assistant,” in Proc. 2024 10th Int. Conf. Advanced Computing and Communication Systems (ICACCS), Thrissur, India, 2024, pp. 1–6. doi: 10.1109/ICACCS60874.2024.10717137.

[2] M. Kulkarni, A. Chivate, M. Chitale, P. Chopade, S. Chitpur, and S. Deshmukh, ”Smart Assistive Navigator for the Blind using Image Processing,” in Proc. 2023 Int. Conf. Sustainable Computing and Smart Systems (ICSCSS), Pune, India, 2023, pp. 1–6. doi: 10.1109/ICSCSS57650.2023.10169767.

[3] D. Munteanu and R. Ionel, ”Voice-Controlled Smart Assistive Device for Visually Impaired Individuals,” in Proc. IEEE Int. Conf., Timişoara, Romania, 2016, pp. 1–6.

[4] P. Mohanraj, T. Rajasekar, N. Sivaelango, V. S. Karthickraja, and N. Vignesh, “Wearable Device for Visually Impaired using Deep Learning,” in Proc. 3rd Int. Conf. Applied Artificial Intelligence and Computing (ICAAIC-2024), Coimbatore, India, 2024, pp. 1–6. doi: 10.1109/ICAAIC60222.2024.10575667.

[5] Y. Lin, K. Wang, W. Yi, and S. Lian, ”Deep Learning Based Wearable Assistive System for Visually Impaired People,” IEEE Trans. Human-Machine Syst., vol. 50, no. 6, pp. 507–516, Dec. 2020. doi: 10.1109/THMS.2020.2992195.

[6] M. A. Khan, P. Paul, M. Rashid, M. Hossain, and M. A. R. Ahad, ”An AI-Based Visual Aid With Integrated Reading Assistant for the Completely Blind,” IEEE Trans. Human-Machine Syst., vol. 51, no. 6, pp. 580–589, Dec. 2021. doi: 10.1109/THMS.2021.3077972.

[7] J. Ai, M. Hu, G. Li, G. Zhai, J. Zhang, Q. Li, and W. Q. Sun, ”Wearable Visually Assistive Device for Blind People to Appreciate Real-world Scene and Screen Image,” in Proc. 2020 IEEE Int. Conf. Visual Communications and Image Processing (VCIP), Shenzhen, China, Dec. 2020, pp. 1–4. doi: 10.1109/VCIP49819.2020.9301778.

[8] R. Král, P. Jacko and T. Vince, ”Low-Cost Multifunctional Assistive Device for Visually Impaired Individuals,” in IEEE Access, vol. 13, pp. 56326-56337, 2025, doi: 10.1109/ACCESS.2025.3554366.

[9] Y. Gao, D. Wu, J. Song, et al., ”A wearable obstacle avoidance device for visually impaired individuals with cross-modal learning,” Nat. Commun., vol. 16, p. 2857, 2025, doi: 10.1038/s41467-025-58085-x.

[10] A. Patil, Y. Bendale, P. Bhangare and S. Patil, ”OdinEye: An AI Based Visual Assistive Device for the Blind and Partially Sighted,” 2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS), Gobichettipalayam, India, 2024, pp. 158-163, doi: 10.1109/ICUIS64676.2024.10866520.

[11] A. N. R. Shree, S. R C, Shankaramma, R. K and P. B A, ”IV Smart: Empowering the Visually Impaired using AI-Driven Assistive Technology,” 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2024, pp. 973-977, doi: 10.1109/ICACRS62842.2024.10841486.

[12] J. Ai et al., ”Wearable Visually Assistive Device for Blind People to Appreciate Real-world Scene and Screen Image,” 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP), Macau, China, 2020, pp. 258-258, doi: 10.1109/VCIP49819.2020.9301814.

[13] N. Tyagi, D. Sharma, J. Singh, B. Sharma and S. Narang, ”Assistive Navigation System for Visually Impaired and Blind People: A Review,” 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV), Gandhinagar, India, 2021, pp. 1-5, doi: 10.1109/AIMV53313.2021.9670951.

[14] D. Munteanu and R. Ionel, ”Voice-controlled smart assistive device for visually impaired individuals,” 2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 2016, pp. 186-190, doi: 10.1109/ISETC.2016.7781087.

[15] N. Bhati, V. C. Samsani, R. Khareta, T. Vashisth, S. Sharma and V. Sugumaran, ”CAPture: A Vision Assistive Cap for People with Visual Impairment,” 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2021, pp. 692-697, doi: 10.1109/SPIN52536.2021.9565940.