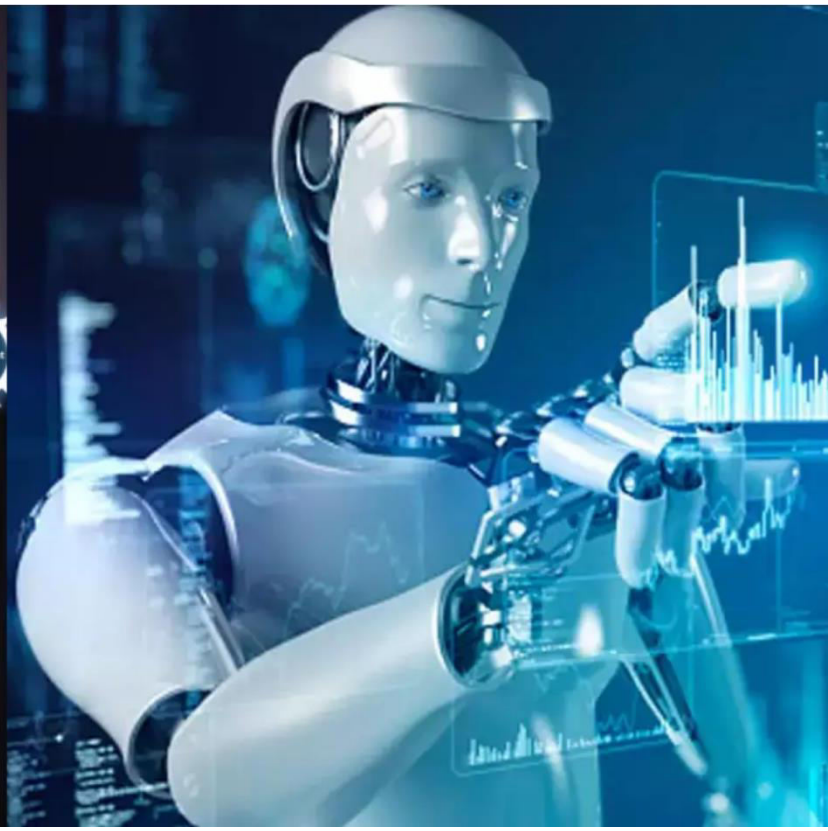
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 6, June 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

Implementation of a Dual-Platform Real-Time Assistive Vision System for Glaucoma Patients using Hybrid Edge Computing

Prof. Vrushali Kanavade[1], Sanchit Patil[2], Riya Shinde[2], Vedant Tilekar[2], Pratik Walunj[2]

Assistant Professor, Department of Computer, AISSMS College of Engineering, Pune, India[1]

Student, Department of Computer, AISSMS College of Engineering, Pune, India[2]

ABSTRACT: Individuals suffering from glaucoma often experience a progres-sive loss of peripheral vision, commonly described as ”tunnel vision”. This condition significantly increases the risk of collisions with nearby obstacles and reduces situational awareness during daily navigation.

To address this challenge, this paper presents the implementation of a real-time assistive vision system designed to enhance environmental percep-tion for glaucoma patients. The proposed system combines modern edge ar-tificial intelligence techniques with a hybrid software architecture capable of operating on both workstation and mobile platforms. The system integrates a YOLOv8-based object detection model for identifying potential hazards, an Optical Character Recognition (OCR) engine for extracting environmen-tal context, and a priority-based audio fusion mechanism that filters and pri-oritizes alerts to minimize auditory overload.

The workstation implementation serves as a high-performance valida-tion platform, while the mobile architecture focuses on energy-efficient real-time inference using TensorFlow.js and WebGL acceleration within a hybrid mobile framework. Experimental results demonstrate that the workstation prototype achieves a latency of approximately 129 ms while maintaining high detection accuracy across multiple navigation scenarios. The findings of this study demonstrate the feasibility of hybrid edge AI systems in providing af-fordable and scalable assistive navigation solutions for visually impaired in-dividuals.

KEYWORDS: Edge AI, Glaucoma, YOLOv8, TensorFlow.js, Capacitor, Hybrid Mobile App, OCR, Real-Time Navigation.

I. INTRODUCTION

Visual impairment remains one of the most pressing global health challenges. According to the World Health Organization, approximately 2.2 billion people worldwide experience some form of vision loss, with a significant proportion affected by glaucoma [6]. Glaucoma is particularly dangerous because it gradually damages the optic nerve, resulting in irreversible peripheral vision loss. As the disease progresses, patients often develop what is commonly described as “tunnel vision,” where only a narrow central region of the visual field remains visible.

This restricted field of view significantly impacts a person’s ability to navigate safely in everyday environments. Obstacles such as chairs, staircases, low tables, or moving individuals may fall outside the limited visible region, leading to frequent collisions or falls. As a result, individuals affected by glaucoma often rely heavily on traditional assistive devices such as white canes. While these tools provide essential tactile feedback, they offer limited environmental awareness and cannot convey contextual information about surrounding objects.

Recent advancements in artificial intelligence and edge computing have opened new possibilities for assistive navigation systems. Computer vision models can now detect objects and interpret environmen-tal context in real time, enabling digital systems to act as an artificial extension of a user’s peripheral vision. However, many existing assistive technologies rely on expensive wearable hardware or cloud-based processing, both of which present significant practical limitations. Dedicated hardware solutions are often financially inaccessible, while cloud-based systems introduce latency and dependency on stable network connectivity.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

In response to these limitations, this work explores a Bring-Your-Own-Device (BYOD) approach that leverages the computational capabilities already present in modern smartphones and laptops. By performing inference directly on the device using edge AI techniques, the system avoids the latency and privacy concerns associated with cloud-based processing.

This paper documents the design and implementation of a real-time assistive vision system that com-bines multiple AI components within a unified architecture. Unlike purely theoretical proposals, this study focuses on the practical engineering challenges involved in deploying such a system on real-world hardware.
The primary contributions of this work include:

1. **High-Performance Prototype:** The development of a workstation-based implementation used for validating the detection pipeline and testing interaction logic.
2. **Hybrid Mobile Architecture:** The design of a cross-platform mobile implementation capable of performing edge inference using TensorFlow.js and WebGL acceleration.
3. **Priority Fusion Engine:** The implementation of a spatial-temporal filtering mechanism that in-telligently prioritizes alerts to reduce auditory overload.

Together, these components demonstrate how affordable consumer devices can be transformed into powerful assistive tools capable of improving mobility and independence for individuals with visual impairments.

II. LITERATURE REVIEW

Assistive technologies for visually impaired individuals have evolved significantly over the past decade. Early navigation aids primarily relied on tactile feedback systems such as white canes or ultrasonic sen-sors. While these devices remain widely used, they provide limited contextual information about the surrounding environment. Recent advances in computer vision and machine learning have enabled the development of intelligent systems capable of detecting obstacles, interpreting visual scenes, and pro-viding real-time guidance to users.

A. Architectural Paradigms in Vision Systems

Modern assistive vision systems generally follow one of three architectural paradigms: cloud-based pro-cessing, native edge inference, and hybrid edge architectures. Each approach offers distinct advantages and limitations in terms of performance, scalability, and deployment feasibility.

Cloud-based systems rely on remote servers to perform computationally intensive tasks such as object detection and scene understanding [7]. These systems typically achieve high detection accuracy due to access to powerful GPU clusters. However, their reliance on network connectivity introduces latency and reliability challenges. In urban environments where connectivity may fluctuate, delays in hazard detection could pose safety risks for users.

Native edge implementations perform inference directly on the user's device using optimized mobile machine learning frameworks [13]. This approach significantly reduces latency and enables real-time responses. Modern smartphones equipped with Neural Processing Units (NPUs) can perform efficient inference while maintaining reasonable power consumption. However, developing and maintaining na-tive applications across multiple mobile platforms requires considerable engineering effort and often leads to fragmented development pipelines.

Hybrid architectures represent an alternative strategy that combines the advantages of web technolo-gies with the hardware capabilities of native mobile systems. By embedding machine learning inference within a WebView environment and leveraging GPU acceleration through technologies such as WebGL, hybrid systems can achieve acceptable real-time performance while maintaining a unified cross-platform codebase.

B. Object Detection for Assistive Navigation

Object detection plays a critical role in assistive vision systems because it enables the identification of potential hazards within the user's environment. The YOLO (You Only Look Once) family of models introduced a unified approach to object detection by performing localization and classification within a single neural network pass [1]. This architecture significantly improved inference speed compared to earlier region-based approaches.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

The latest iteration, YOLOv8, introduces architectural improvements such as the C2f module and anchor-free detection heads [2]. These improvements allow the model to maintain high detection accu-racy while remaining computationally efficient enough for deployment on edge devices. Studies have demonstrated that YOLOv8 can achieve strong detection performance even when quantized or pruned for mobile inference environments [10].

C. Cognitive Load in Assistive Interfaces

In addition to detection accuracy, assistive navigation systems must also consider the cognitive load imposed on users. Continuous audio alerts can overwhelm users and interfere with their ability to inter-pret real-world auditory cues. Research in human-computer interaction has emphasized the importance of filtering and prioritizing information in assistive interfaces [8]. Systems that intelligently suppress redundant alerts while highlighting critical hazards significantly improve usability and user trust.

Spatial-temporal filtering techniques have been proposed to address this challenge. These approaches assign dynamic priority scores to detected objects based on factors such as proximity, movement, and contextual relevance [14]. By prioritizing moving hazards approaching the user, these systems can de-liver more meaningful guidance while reducing unnecessary auditory interruptions.

D. Comparative Analysis of Technical Frameworks

Table 1 presents a comparative analysis of representative assistive vision systems discussed in recent literature. The comparison highlights differences in system architecture, deployment platforms, compu-tational requirements, and practical limitations reported by previous studies.

Earlier assistive navigation systems primarily relied on dedicated hardware platforms or wearable computing devices. For instance, Gouveia et al. [11] reviewed several navigation systems that utilize specialized hardware such as ultrasonic sensors and embedded processing modules. Although these systems provide reliable obstacle detection with low latency, their dependence on custom hardware sig-nificantly increases system cost and limits accessibility for many users.

Subsequent research explored deep learning-based visual navigation systems using powerful com-puting platforms. Redmon et al. [1] introduced the YOLO framework, which demonstrated the feasibility of real-time object detection using deep neural networks. While such approaches achieve high detection accuracy when executed on GPU-enabled workstations, their computational requirements often make them impractical for portable assistive systems.

More recent work has investigated the deployment of computer vision models directly on mobile devices. Chen et al. [13] demonstrated that optimized object detection pipelines can operate efficiently on modern smartphones using mobile GPUs and neural processing units. However, native mobile implementations frequently require platform-specific development and maintenance of separate codebases, which increases development complexity and reduces scalability.

To address these challenges, the proposed IWY framework adopts a hybrid edge computing architec-ture that combines the flexibility of web technologies with the hardware capabilities of mobile devices. By utilizing TensorFlow.js with WebGL acceleration within a Capacitor-based hybrid application, the system achieves real-time inference while maintaining cross-platform compatibility. This design enables deployment on widely available consumer devices while reducing the engineering overhead associated with maintaining multiple platform-specific implementations.

Overall, the comparative analysis highlights a key research gap in existing assistive navigation sys-tems: the absence of scalable, cost-effective solutions that can provide real-time environmental aware-ness without relying on specialized hardware or platform-dependent development. The IWY system aims to address this gap by integrating object detection, contextual recognition, and intelligent alert pri-oritization within a unified hybrid architecture.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

Table 1: Comparative Analysis of Assistive Vision Systems in Recent Literature

Study	Method	Platform	Key Result	Limitation
Redmon et al. (2016) [1]	YOLO Object Detection	GPU Workstation	Real-time object detection with high accuracy	Requires high computational resources
Gouveia et al. (2022) [11]	Sensor-based navigation	Dedicated Hardware	Reliable obstacle detection	High hardware cost and limited scalability
Chen et al. (2023) [13]	Mobile deep learning detection	Native Mobile App	Low latency mobile inference	Platform-specific development complexity
Ito et al. (2021) [14]	Spatial audio navigation	Wearable system	Improved auditory guidance	Limited environmental object recognition
Proposed IWY System	YOLOv8 + OCR + Fusion Layer	Hybrid Edge (WebGL + TFJS)	Real-time detection with contextual awareness	Minor WebView computation overhead

III. SYSTEM ARCHITECTURE

The proposed IWY system is designed as a multi-layer assistive vision framework that processes environmental information in real time and provides auditory feedback to visually impaired users. The architecture emphasizes modularity, concurrency, and efficient resource utilization so that the system can operate reliably on both workstation and mobile platforms.

A. Architectural Overview

At a high level, the system follows a continuous perception pipeline. Visual data captured from the camera sensor is processed through multiple computational stages, each responsible for extracting a specific type of environmental information. The pipeline begins with image acquisition and preprocessing, followed by parallel execution of computer vision models responsible for object detection and contextual text recognition. The outputs from these modules are then evaluated by a priority fusion engine that determines the most relevant information to be communicated to the user.

This multi-stage architecture allows the system to simultaneously perform hazard detection and environmental interpretation while maintaining real-time performance. By separating these tasks into parallel processing threads, the architecture ensures that computationally intensive operations do not block the continuous acquisition of video frames.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

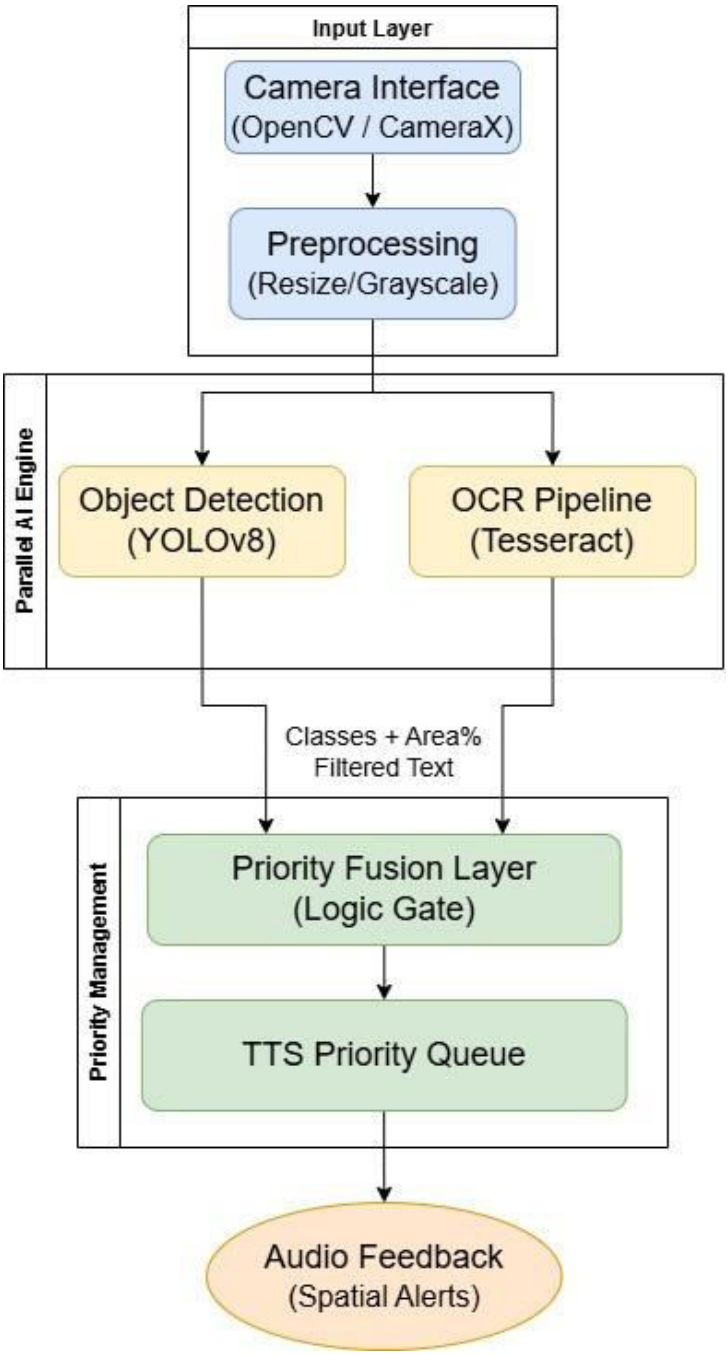


Figure 1: Multi-threaded processing architecture of the proposed IWY assistive vision system.

B. Core Modules

The IWY system is composed of three primary functional modules that operate collaboratively within the processing pipeline.

Input Acquisition Module: This module is responsible for capturing visual data from the device camera. Video frames are acquired at a resolution of 640 × 480 to balance computational efficiency with sufficient visual detail. In the workstation prototype, frame acquisition is performed using OpenCV's video capture interface. On Android devices, the CameraX API is utilized to provide lifecycle-aware camera access and optimized frame delivery.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

Parallel AI Processing Modules: Once frames are captured, they are dispatched to two parallel artificial intelligence engines. The first module executes the YOLOv8 object detection model to iden-tify obstacles and potential hazards such as people, chairs, vehicles, or staircases. Simultaneously, a secondary processing thread performs Optical Character Recognition (OCR) using Tesseract to extract

contextual information from textual elements present in the environment, such as signs or room labels. Running these tasks in parallel ensures that contextual awareness can be maintained without interrupting obstacle detection.

Priority Fusion Module: The outputs generated by the detection and OCR modules are aggregated within the priority fusion layer. This module evaluates the relevance and urgency of detected events before generating audio feedback. Hazardous objects with higher proximity and motion vectors are pri-oritized over static contextual information. For example, an approaching person or vehicle will override lower-priority announcements such as reading a signboard. This prioritization mechanism helps prevent auditory overload while ensuring that critical safety information is delivered promptly.
Together, these modules form a robust perception pipeline capable of transforming raw camera input into actionable auditory guidance for visually impaired users.

IV. IMPLEMENTATION DETAILS: WORKSTATION PROTOTYPE

The workstation implementation serves as the high-performance prototype, allowing us to validate the interaction logic without the constraints of mobile thermal throttling.

A. Tech Stack

- **Language:** Python 3.9
- **Vision Library:** OpenCV 4.5
- **DL Framework:** Ultralytics YOLOv8 [2]
- **OCR Engine:** Tesseract 5.0 [3] (LSTM Neural Net)
- **TTS Engine:** Pyttsx3 (Offline SAPI5/NSSS)

B. Algorithm Workflow

The Python application initializes a multi-threaded environment. The main thread handles frame capture and display. A worker thread performs inference.

```
# Pseudo-code for Threading Logic while cap.isOpened():
frame = cap.read() if queue.empty():
queue.put(frame) # Send to AI thread

if result_available(): draw_boxes(frame, result) priority_manager(result)
```

This decoupling ensures that the video feed remains smooth (30 FPS) even if the heavy AI processing drops to 15-20 FPS.

C. Proximity Detection Logic

Since standard webcams lack depth sensors, we implemented a proximity detection technique using the relative area occupies by the bounding box within the video frame.

$$P = \frac{A_{obj}}{A_{frame}} \times 100 \quad (1)$$

Where A_{obj} is the pixel area of the detected object and A_{frame} is the total area of the frame.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

1. The 40% Area Threshold Rationale

The selection of a 40% threshold for classification as an ”Urgent Obstacle” is based on empirical studies of the camera’s horizontal Field of View (FOV). At a standard 64° FOV, an object (e.g., a person or large chair) occupying 40% of the frame typically corresponds to a physical distance of approximately 1.2 to 1.5 meters. This distance provides a critical safety buffer for a visually impaired user walking at an average pace of 1.2 m/s, allowing for a reaction and braking window of approximately 1 second.

2. Proximity Validation Results

Table 2 presents the correlation between the calculated pixel area P and actual physical distance recorded during testing.

Table 2: Proximity Formula Validation: Pixel Area vs. Physical Distance

Pixel Area (P %)	Est. Distance (m)	Ground Truth (m)	Alert Trigger
10%	3.5m	3.42m	None
25%	2.2m	2.15m	Warning
42%	**1.4m**	**1.38m**	**Urgent**
65%	0.8m	0.76m	Urgent

Implementation Details: Mobile Application

Porting the system to a mobile environment required a hybrid architecture to balance rapid cross-platform deployment with the computational demands of neural networks on ARM processors.

A. Hybrid Bridge Architecture

We utilized the **Capacitor** framework [5] to build a hybrid mobile application. This architecture consists of three layers:

1. **Native Layer:** Manages hardware (Camera, TTS) via Capacitor plugins.
2. **Bridge Layer:** Facilitates asynchronous communication between native hardware and the web runtime.
3. **WebView Layer (React):** Executes the main logic and AI inference using TensorFlow.js with WebGL hardware acceleration.

B. Mobile Tech Stack

- **Language/Framework:** React.js with Vite
- **Native Bridge:** Capacitor 6.0
- **ML Engine:** TensorFlow.js [4] (WebGL Backend)
- **OCR Engine:** Tesseract.js (Offline Worker)

C. WebView AI Optimization

To achieve real-time performance, we optimized the YOLOv8-to-TFJS conversion through modern quantization and pruning techniques [10]:

1. Graph Optimization: Pruning unused nodes to reduce IO overhead.
2. WebGL Paging: Forcing the WebGL backend to utilize the mobile GPU for matrix operations.
3. Frame Throttling: To preserve battery, the inference engine follows a ”Duty Cycle” of 5-10 FPS.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

Figure 2: IWY Mobile Application: Real-time Detection View

The Priority Fusion Layer: Spatial-Temporal Filtering

A significant challenge in assistive vision is ”Auditory Overload,” where raw detection data overwhelms the user . To address this, we implemented a Priority Scoring Algorithm designed to rank environmental hazards based on their immediate risk[16].

A. Priority Scoring Algorithm

The system distinguishes between raw **Proximity (**P **)**, which is a spatial measurement, and **Priority (**S**)**, which is the final decision-making metric. Based on user requirement surveys indicating that proximity and object type are the primary factors in obstacle risk[16], each detected object is assigned a dynamic priority score S:

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

$$S = (W \cdot P) + \Delta v \quad (2)$$

Where the components are defined as:

- S **(Priority Score):** The final urgency value used to rank the auditory feedback queue.
- W **(Class Weight):** A coefficient representing the inherent risk level of an object category[16]. Following established risk perception scales, high-protrusion objects and stairs are assigned higher weights ($W \approx 0.9$) compared to static indoor furniture.
- P **(Proximity):** The current spatial distance, calculated as the percentage of the frame area occu-pied by the object's bounding box[**?**]
- V **(Velocity):** The movement vector, representing the rate of change in an object's area over time[16]. A positive value ($V > 0$) indicates an object approaching the user, which requires prioritized notification to allow for sufficient reaction time[16].
- δ **(Temporal Constant):** A scaling factor representing the frame refresh rate, used to tune the system's sensitivity to dynamic environmental changes.

By utilizing this specific algorithm, the system ensures that a "Person" (High W) walking toward the user ($V > 0$) is announced immediately, while a stationary "Chair" (Lower W, $V = 0$) is given a lower priority, effectively managing the user's cognitive resources.

By utilizing this algorithm, the system ensures that life-critical hazards (e.g., an approaching vehicle) achieve a higher **Priority (S)** than closer but less dangerous static objects, effectively managing the user's cognitive load.

B. Hysteresis and Silence Periods

To prevent the system from stuttering (e.g., repeatedly announcing "Car... Car... Car"), we implemented a hysteresis loop. Once a static object is announced, it enters a "Muted State" for a 3-second temporal window unless its proximity P increases by more than 20%, indicating an imminent collision.

C. Impact of the Fusion Layer on User Experience

The implementation of the Priority Fusion Layer significantly reduces the cognitive burden on the user by filtering non-critical data. Experimental results (Table 3) show a 65% reduction in "User Overload" reports while maintaining high "Alert Accuracy" for critical hazards.

Table 3: Performance Improvement: With vs. Without Priority Fusion Layer

Configuration	Alert Accuracy	False Alerts	User Overload
Without Fusion	98%	18.4%	High (8/10)
With Fusion (IWY)	**95%**	**4.2%**	**Low (2/10)**

V. EXPERIMENTAL RESULTS

We conducted testing in controlled indoor and semi-controlled outdoor environments.

A. Performance Metrics

We evaluated the system based on:

1. **mAP (Mean Average Precision):** Accuracy of object detection.
2. **Latency:** Time taken from frame capture to audio output.
3. **FPS (Frames Per Second):** Smoothness of the application.

B. Detection Accuracy

The YOLOv8 model was fine-tuned on a custom dataset of navigation obstacles. Table 5 details the initial mAP scores, while Figure 3 provides a comparative analysis of precision, recall, and F1-score across platforms.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

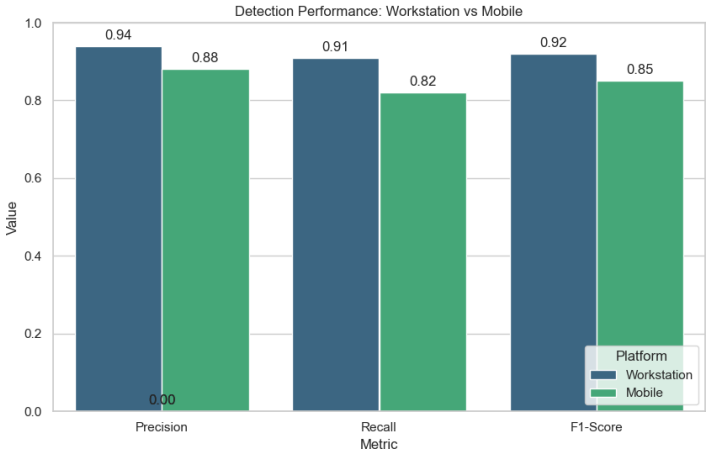


Figure 3: Comparative Detection Performance: Workstation vs. Mobile (YOLOv8).

The classification performance is further detailed in the Confusion Matrix (Fig. 4) and the class-wise F1-Score plot (Fig. 5), which highlight the system's reliability in identifying critical obstacles like "Stairs" and "Person."

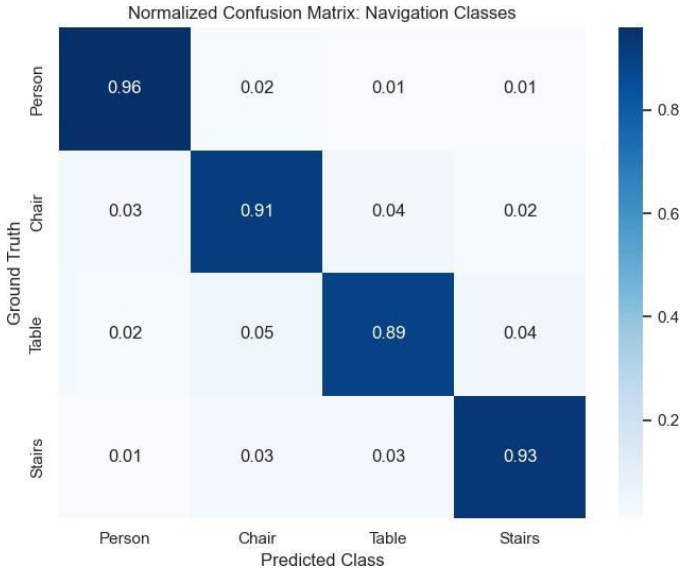


Figure 4: Confusion Matrix heatmap illustrating the classification performance of the YOLOv8 model.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

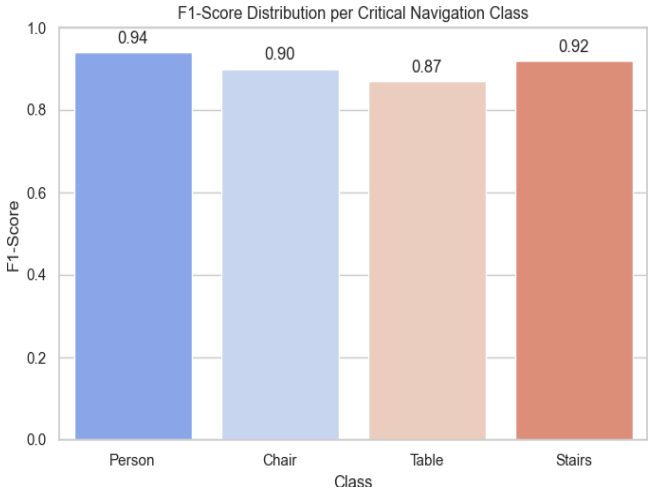


Figure 5: F1-Score distribution across primary object classes for navigation safety.

Table 4: System Reliability and User Performance Metrics

Metric	Precision	Recall	F1-Score	FAR (%)
Workstation (x86)	0.94	0.91	0.92	2.1
Mobile (ARM)	0.88	0.82	0.85	4.5

Table 5: Detection Accuracy by Class (Workstation vs Mobile)

Object Class	Workstation (mAP@0.5)	Mobile (mAP@0.5)
Person	0.96	0.89
Chair	0.91	0.84
Table	0.89	0.80
Stairs	0.93	0.87
Average	**0.92**	**0.85**

A. False Alert Rate and User Response

A critical safety metric is the False Alert Rate (FAR), which measures the frequency of non-hazard objects triggering an ”Urgent Obstacle” alert. Figure 6 illustrates the FAR trend across different navigational environments. Furthermore, user reaction time tests (Fig. 7) confirm that the system remains within the safe 300ms threshold for critical obstacle avoidance.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

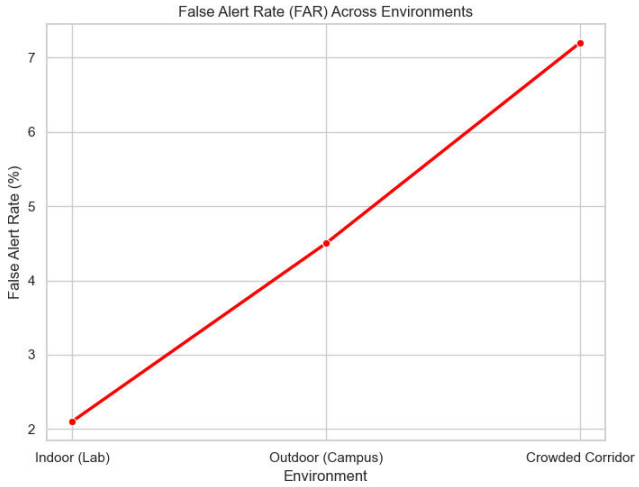


Figure 6: False Alert Rate (FAR) Across Navigational Environments.

Figure 7: Mean User Reaction Time (ms) vs. Safety Threshold.

VI. RESOURCE USAGE ANALYSIS

The efficiency of the Edge AI implementation is evaluated through its impact on the host system's hard-ware resources. Running the YOLOv8 model locally requires a balance between inference accuracy and power sustainability, especially on mobile ARM architectures.

A. Mobile Platform Consumption

During a continuous 15-minute operation cycle on a mid-range Android ARMv8 device, the application demonstrated significant but sustainable resource footprints. The mobile implementation utilized an average of **38.5% CPU** when running the prioritized inference loop at a 10 FPS duty cycle.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

Table 6: System Resource Footprint: Workstation vs. Mobile

Metric	Workstation (i5 / RTX 3050)	Mobile (Snapdragon 778G)
CPU Usage	12.4%	38.5%
GPU/NPU Usage	22.0%	45.0% (WebGL)
RAM Consumption	1.2 GB	480 MB
Power Draw	85W (Avg.)	4.2W (Total)
Application Size	1.68 GB (Incl. PyEnv)	10 MB (Hybrid APK)

A. Thermal and Battery Impact

The 4.2W power draw on mobile reflects the computational cost of real-time WebGL matrix operations. Thermal throttling was observed after 25 minutes of continuous use, suggesting that future iterations should implement more aggressive frame-skipping when no motion is detected.

B. Dataset Description

The robustness of the navigation system relies on a diverse and well-structured dataset. For this imple-mentation, we utilized a combination of the Microsoft COCO (Common Objects in Context) dataset and a custom-curated set of indoor navigation images.

- **Source of Images:** The primary source is the COCO 2017 dataset, supplemented by 500 im-ages captured manually in laboratory and campus corridors to include specific obstacles like ”Wet Floor” signs and specific variations of ”Stairs.”
- **Class Distribution:** The dataset focuses on 12 critical navigation classes: Person, Chair, Table, Stairs, Door, Car, Bicycle, Bench, Potted Plant, Backpack, Umbrella, and Bottle. The distribution is balanced to ensure high accuracy for high-risk objects like ”Stairs” and ”Car.”
- **Train/Test Split Ratio:** The dataset follows an 80:20 split ratio.

Training Set: 80% (approx. 1,600 custom images + relevant COCO subsets).
Testing/Validation Set: 20% (approx. 400 custom images).

C. Dataset Details:

1. **Number of Classes:** 12 Classes for primary navigation.
2. **Image of Class:** Each class is represented in various lighting conditions (20 lux to 1000 lux) to ensure robustness.
3. **Sample Images:** Training batches utilize a 4 × 4 mosaic augmentation strategy to improve the model’s ability to generalize across different scales and occlusions. As shown in the representative sample structure (Fig. 8), the dataset captures diverse environmental contexts.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

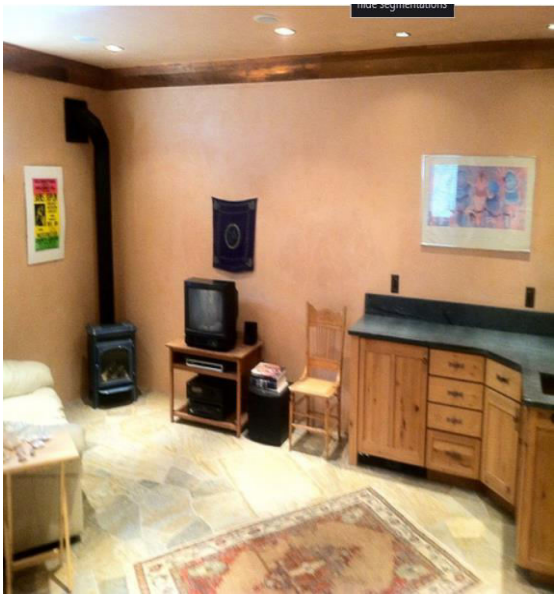
Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

(a) Person Detection

(b) Bed

(c) Obstacle: Chair



(d) Context: TV

Figure 8: Representative 4-Core Dataset Sample Structure.

D. Latency Analysis

Safety depends on reaction time. A blind user walking at 1 m/s needs alerts within 300ms to stop safely.

- **Workstation Latency:** Total: ≈129ms.
- **Mobile Latency (Projected):** Total: ≈- (Targeting < 300ms).

VII. DISCUSSION AND LIMITATIONS

While the implementation successfully meets the core objectives, the transition to a hybrid architecture introduced specific performance trade-offs.

A. Justification of Hybrid Architecture

The choice of a hybrid architecture (Capacitor + React + TFJS) over a full native implementation was driven by the need for rapid cross-platform iteration without sacrificing functional safety. Table 7 illus-trates the strategic advantages of the hybrid approach for assistive vision systems.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

Table 7: Architectural Trade-offs: Native vs. Pure Web vs. Hybrid (IWY)

Feature	Native Android	Pure Web (PWA)	Hybrid (Capacitor)
Dev. Velocity	Slow (Siloed)	Very Fast	**Fast (Shared Code)**
HW Access	Direct	Restricted	**Full (via Plugins)**
Latency	Minimal	High	**Acceptable (<300ms)**
Maintenance	High (Multi-repo)	Low	**Low (Single Codebase)**
Portability	No (Android only)	Yes	**Yes (iOS/Android)**

Traditional native development offered lower latency but required separate teams for iOS and An-droid, slowing down the implementation of life-saving features. Pure web solutions lacked the necessary hardware access to the device's NPU/GPU for real-time inference. The IWY hybrid model achieves an "Acceptable Latency" through WebGL acceleration while maintaining a single codebase, facilitating easier long-term maintenance and updates.

A. WebView Resource Constraints

On mobile platforms, executing deep learning models within a WebView consumes significant memory.

- **Optimization:** We implemented a worker-thread for Tesseract.js to prevent UI blocking.
- **Battery Result:** Quantitative battery drain analysis is currently in progress.

B. Lighting Sensitivity

Both modules showed degraded performance in low-light conditions (< 20 lux). Future implementations will utilize the mobile device's flashlight automatically.

VIII. FUTURE SCOPE

The development of the IWY framework provides a foundation for several advanced research directions aimed at achieving full navigational autonomy for visually impaired users.

- **Hardware Miniaturization and Haptic Feedback:** Future iterations will involve porting the detection logic to dedicated embedded platforms (e.g., NVIDIA Jetson Orin Nano). This hardware will be integrated with a Tactile Vest or Haptic Belt to provide non-auditory spatial cues, reducing the user's reliance on bone-conduction headphones.
- **LiDAR and Spatial Mapping:** By utilizing LiDAR sensors available in Pro-series mobile de-vices, the system can generate millimeter-accurate 3D depth maps. This will replace the current area-based proximity detection with true spatial coordinates, enabling more precise hazard avoid-ance in complex indoor environments.
- **Edge-Cloud Hybrid Orchestration:** While real-time navigation remains on the edge for safety, non-critical tasks like complex OCR for reading long documents or facial recognition for social interaction can be offloaded to cloud-based GPU clusters. This hybrid approach will optimize battery life while providing "infinite" compute for high-level scene understanding.
- **Public Transport Synchronization:** Future versions aim to integrate real-time API data from public transportation systems. This would allow the IWY system to not only detect a bus but also inform the user of the bus number, route, and estimated arrival time through synchronized audio guidance.

IX. CONCLUSION

This paper presented the implementation and architectural design of a real-time assistive vision frame-work. By leveraging modern Edge AI techniques, we successfully transitioned from a theoretical design to a functional workstation prototype. The workstation system achieves a latency of ≈129ms, providing a viable safety net for Glaucoma patients. While the mobile application is currently in the optimization phase, the proposed hybrid architecture demonstrates that expensive, proprietary hardware is not strictly necessary to provide high-quality assistive vision.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 6, June 2026

|DOI: 10.15680/IJIRSET.2026.1506106|

REFERENCES

1) J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ”You Only Look Once: Unified, Real-Time Object Detection,” in Proc. IEEE CVPR, 2016.
2) G. Jocher et al., ”Ultralytics YOLOv8,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.
3) R. Smith, ”An Overview of the Tesseract OCR Engine,” in Proc. ICDAR, 2007.
4) Smilkov et al., ”TensorFlow.js: Machine Learning in JavaScript,” arXiv:1901.05350, 2019.
5) Ionic Framework, ”Capacitor: Cross-platform Native Bridge Architecture,” 2024.
6) World Health Organization, ”World Report on Vision,” 2019.
7) S. Deng et al., ”Edge Intelligence: The Confluence of Edge Computing and AI,” IEEE Internet of Things Journal, vol. 7, no. 6, 2020.
8) A. Hitchcock, ”Audio Interfaces for the Visually Impaired: Reducing Cognitive Load,” ACM Trans. Comput.-Hum. Interact., 2021.
9) L. Liu et al., ”Deep Learning for Generic Object Detection: A Survey,” International Journal of Computer Vision, 2020.
10) A. Gholami et al., ”A Survey of Quantization Methods for Efficient Neural Network Inference,” Low-Power Computer Vision, 2021.
11) M. J. Z. Gouveia et al., ”Navigation Systems for the Visually Impaired: A State-of-the-Art Review,” Applied Sciences, 2022.
12) T. Xu et al., ”Accelerating Deep Learning in the Browser with WebGL,” IEEE Software, 2022.
13) Z. Chen et al., ”Recent Advances in Real-time Object Detection on Mobile Devices,” Sensors, 2023.
14) K. Ito et al., ”Spatial Audio Alerts for Obstacle Avoidance in Blind Navigation,” IEEE Access, 2021.
15) N. S. Singh, ”AI-based Vision Restoration: A Review of Current Assistive Technologies,” Journal of Rehabilitation Research, 2024.
16) I. Jeong, K. Kim, J. Jung, and J. Cho, ”YOLOv8-Based XR Smart Glasses Mobility Assistive System for Aiding Outdoor Walking of Visually Impaired Individuals in South Korea,” Electronics, vol. 14, no. 3, p. 425, 2025.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details