ECE445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

AI-Based Visual Navigation System for Assistive Smart Glasses

---

Team 45

Mingyan Gao [mg81]

Shengnan Cai [sc129]

Yi Su [yi31]

Junchen He [he58]

Advisor: Weeliat Ong

April 8, 2026

Contents

1. Introduction

1.1 Problem Statement

Visually impaired individuals face significant challenges in independent navigation, lacking reliable, real-time environmental awareness. Critical tasks such as following tactile paving (blind paths), avoiding obstacles, detecting crosswalks, and interpreting traffic signals require continuous situational awareness that is profoundly difficult to achieve without vision.

Existing assistive tools and automated solutions exhibit major shortcomings. Traditional tools provide extremely limited spatial information. More advanced electronic travel aids often overwhelm users with excessive, non-prioritized auditory feedback, and crucially fail to adapt to the user's dynamic movement, such as continuous head movement and camera shaking during a walk. As a result, users are exposed to elevated risks, including collisions with dynamic obstacles (like pedestrians or unexpectedly opened doors), unsafe street crossing decisions, and losing alignment with safe walking paths.

The proposed Intelligent Wearable Vision System addresses these critical gaps by deploying a smart glasses hardware platform integrated with a camera and IMU sensing. Instead of overwhelming the user, the system strictly filters environmental data to detect and notify only regarding critical hazards, such as stairs, obstacles, pedestrians, doors, and trees. Furthermore, by fusing optical flow and IMU data, the system adapts its feedback based on user motion to provide stable, real-time directional guidance, fundamentally bridging the gap between static environmental perception and dynamic human navigation.

1.2 Solution Overview & Visual Aid

The system is deployed as a wearable smart glasses device interacting with a local edge-processing backend. The hardware suite includes an ESP32-CAM for visual data acquisition and an ESP32-IMU for tracking head motion, powered by a 3.7V 1000mAh battery. These modules stream data to a FastAPI server where advanced computer vision algorithms—primarily a YOLO segmentation model—perform real-time blind path segmentation, traffic light recognition, and obstacle detection. [4]

Once objects are identified, the navigation state machine calculates dynamic directional arrows and offsets. This feedback is adaptively stabilized against camera shake and delivered back to the user via the smart glasses, providing an intuitive, prioritized, and non-intrusive assistive experience.

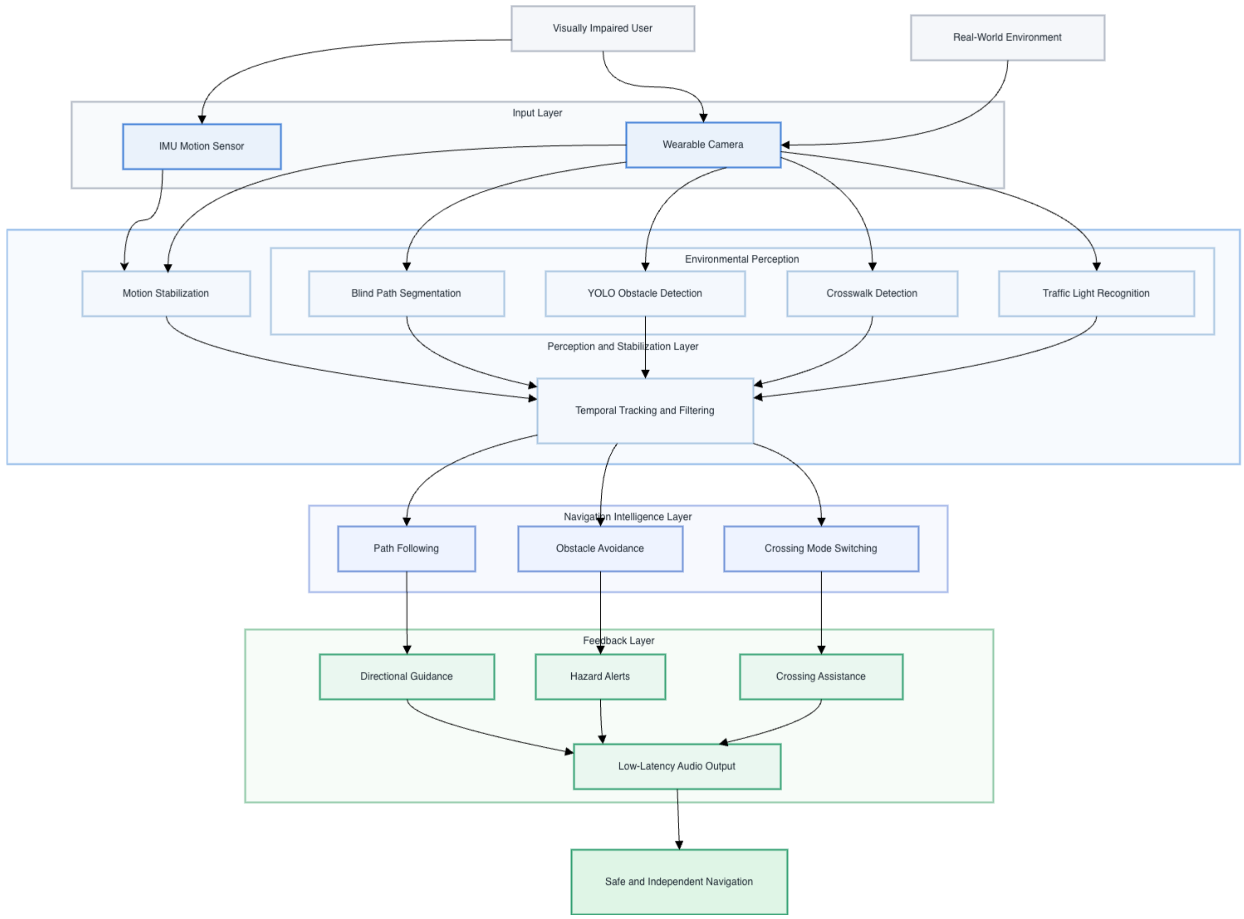

Figure 1: Visual Aid of the Wearable Vision System in its Operational Context.

1.3 High-Level Requirements List

To successfully replace traditional navigation aids and ensure user safety, the system must meet the following three comprehensive requirements:

1. **Perception and Segmentation Accuracy:** The system must reliably identify critical navigation elements in real-world scenarios. It shall achieve an overall obstacle and blind-path detection accuracy of $\geq 90\%$ at distances up to 3 meters, and maintain a traffic light recognition precision of $\geq 85\%$ in varied outdoor lighting conditions.

2. **Real-Time Processing and Responsiveness:** Because the user is constantly in motion, computational lag can cause fatal misdirection. The integrated pipeline—from ESP32-CAM capture to YOLO inference on the FastAPI server—must maintain a sustained processing throughput of ≥ 15 FPS, with an end-to-end latency not exceeding 1000 ms.

3. **Dynamic Stabilization and Navigation Reliability:** To prevent the loss of blind path masks caused by continuous head movement and camera shaking, the system must successfully fuse IMU and optical flow data. It shall consistently output directional guidance offsets with a mask deviation error of ≤ 10

pixels frame-by-frame, ensuring smooth and uninterrupted user feedback.

2. Design

2.1 Block Diagram

The system architecture is divided into three functional layers: the Sensing Layer (Hardware), the Processing Layer (Cloud/Server), and the Feedback Layer (Navigation Logic).

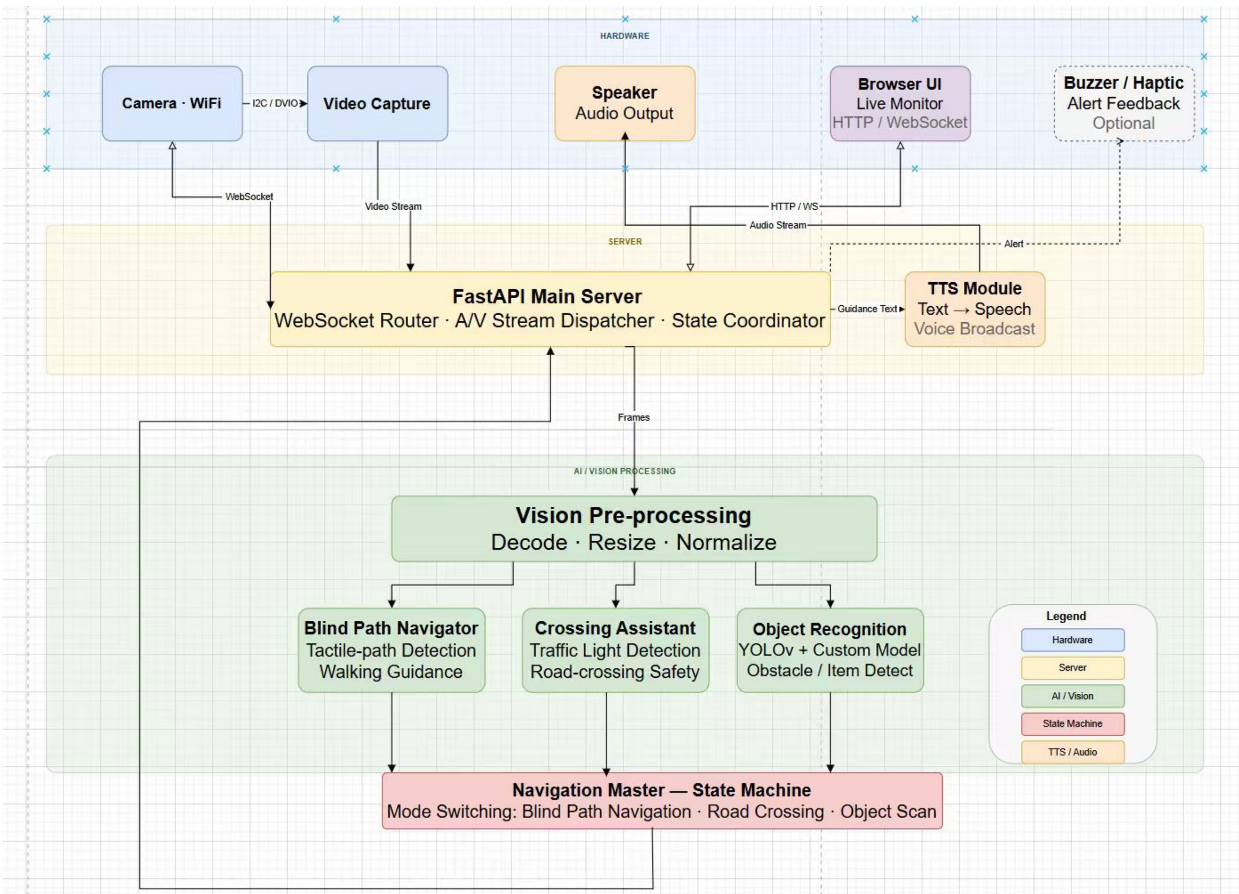

Figure 2: System block diagram illustrating the flow from raw sensor data to stabilized navigation guidance

2.2 Subsystem Descriptions

2.2.1 Sensing & Perception Subsystem

The primary function of this subsystem is to gather multi-modal data from the user's environment. The ESP32-CAM captures a continuous JPEG stream of the path ahead, while the ESP32-IMU records the orientation and angular velocity of the user's head. This dual-sensor approach is critical for compensating for the inherent instability of a wearable camera.

2.2.2 Cloud Computing & Computer Vision Subsystem

The central server acts as the "brain" of the system. It hosts a FastAPI-based WebSocket server to handle concurrent data streams. A YOLO segmentation model processes each frame to generate binary masks for blind paths and bounding boxes for obstacles. This subsystem is designed for high-throughput to minimize the delay between perception and action.

2.2.3 Navigation Logic & Stabilization Subsystem

This subsystem bridges the gap between raw detection and user guidance. It employs a navigation state machine to interpret environmental contexts (e.g., standard walking vs. crossing a street). Crucially, it integrates IMU data with optical flow to stabilize detection masks that would otherwise "jitter" due to head movement, ensuring that the directional guidance remains steady.

2.3 Subsystem Requirements & Verification

Subsystem Requirements	Verification
Sensing & Perception Subsystem	
1. **Vision Integration:** The integrated camera pipeline must capture and transmit 320×240 resolution frames at a sustained rate of 10 ± 3 FPS over a 2.4GHz Wi-Fi connection to ensure continuous perception.	1. **Procedure:** Power the device and connect to the local test network. Run a Python script on the server to log the timestamp of each received frame for 10 minutes. **Success Criterion:** At least 90% of frame intervals are within 500 ms, and no single gap exceeds 3000 ms.
Cloud Computing & Computer Vision Subsystem	
1. **System Latency:** The end-to-end processing latency—from the physical event occurring in front of the camera to the generation of the navigation command—must be $\leq$ 800 ms under typical operating conditions.	1. **Procedure:** Place a programmable LED in the camera's FOV. Trigger the LED and a hardware timer simultaneously. Stop the timer when the server outputs the detection signal. Repeat this test 30 times. **Success Criterion:** At least 80% of the recorded latencies are $\leq$ 800 ms, and the maximum latency does not exceed 1200 ms.
2. **Algorithmic Accuracy:** The custom-trained YOLO segmentation model must achieve a mean Intersection over Union (mIoU) of $\geq$ 65% for blind path segmentation and $\geq$ 60% for dynamic obstacles.	2. **Procedure:** Evaluate the model offline using **self-recorded video datasets** of various campus walking paths. Compare the model's output masks against manually annotated ground-truth labels. **Success Criterion:** The evaluation script outputs an overall mIoU of $\geq$ 65% for blind paths and $\geq$ 60% for obstacles, with zero false negatives for obstacles within 2 meters.
Navigation Logic & Stabilization Subsystem	
1. **Dynamic Stabilization:** The sensor fusion algorithm must maintain the calculated directional guidance offset within 15% of the true path center when subjected to continuous yaw rotations of up to $5^\circ/s$.	1. **Procedure: Analyze frame-by-frame stability on self-recorded videos** containing intentional head movements. Compare the stabilized center point coordinates against the baseline (stationary) center point during camera rotation. **Success Criterion:** Data analysis demonstrates that the stabilized center deviation does not exceed 15% from the baseline center in at least 85% of the frames recorded during the rotation period.

2.4 Supporting Material

This section provides visual and quantitative evidence to effectively communicate the technical depth and integration logic of the Intelligent Wearable Vision System.

2.4.1 System Processing Pipeline

The following flow logic illustrates the modular transition from raw sensor acquisition to the final stabilized feedback. This hierarchical structure ensures that the latency-critical perception tasks are prioritized before the stabilization and navigation logic.

- **Data Ingestion Layer:** Synchronizes 320×240 JPEG frames via WebSocket and 6-axis IMU quaternions via UART/JSON.

- **Perception Engine:** Utilizes a YOLO-based segmentation model to isolate the blind path geometry and define obstacle bounding boxes.
- **Stabilization & Fusion:** Mathematically shifts the visual coordinate system based on IMU-derived head rotation angles to compensate for camera shake.
- **Instruction Logic:** Translates the final stabilized pixel offsets into discrete navigation commands.

2.4.2 Key Design Parameters

Table 1 summarizes the quantitative design specifications that enable the system to meet the high-level requirements. These parameters are chosen to balance real-time responsiveness with wearable power constraints.

Technical Parameter	Design Specification
Communication Protocol	Hybrid WebSocket (Binary for Video) and REST/JSON (for Control)
Target System Latency	$\leq$ 1.0 s (End-to-end processing delay)
IMU Sampling Rate	100 Hz (Fixed interval for high-fidelity pose estimation)
Power Architecture	3.7V 1000mAh Li-Po battery with regulated 3.3V/5V rails
Stabilization Tolerance	$\leq$ 5% of total FOV deviation during $3°/s$ head rotation

Table 1: System technical specifications and design constraints.

2.4.3 Hardware Integration Blueprint

The physical design focuses on ergonomic weight distribution and sensor alignment. The ESP32-CAM is bridge-mounted to align the camera's optical axis with the user's forward gaze, while the IMU is temple-mounted to capture pure rotational data of the skull. This spatial configuration is critical for the accuracy of the mathematical stabilization model described in the Tolerance Analysis.

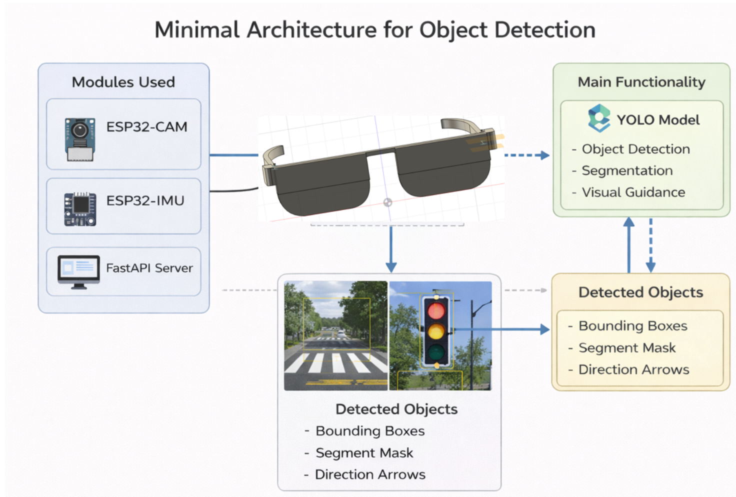

Figure 3: Hardware integration layout showing the spatial distribution of sensing and power modules.

2.5 Tolerance Analysis

The most critical challenge identified in this project is the **inaccuracy of navigation guidance caused by significant system processing latency (end-to-end delay)**. In a real-time assistive device, there is an inherent time delay (t_{delay}) between the moment the camera captures an image and the moment the server completes the YOLO inference and delivers the corresponding feedback.

When a user walks, their head rotates at an average angular velocity (ω). During this substantial processing delay (t_{delay}), the user's head continues to rotate, causing a spatial discrepancy between the captured frame's orientation and the user's actual current orientation. This results in a horizontal angular shift ($\Delta\theta$) of the detected path center relative to the user's real-time perspective. This shift is expressed as a percentage of the camera's total Field of View (FOV):

$$\text{Error}(\%) = \frac{\omega \cdot t_{delay}}{\text{FOV}_{horizontal}} \times 100\%$$

Assuming a steady walking head turn velocity of $3^\circ/s$, a system latency of 1 second ($t_{delay} = 1.0s$), and a camera FOV of 60°:

$$\text{Error}(\%) = \frac{3^\circ/s \cdot 1.0s}{60^\circ} \times 100\% = 5\%$$

Although this latent displacement results in a 5% offset of the visual field, it is important to note that the **recognition probability of the vision model remains exceptionally high**. By offloading computation to a GPU-accelerated FastAPI server, the YOLO model utilizes deep neural networks that maintain a detection confidence score of $\geq 90\%$ for critical markers like blind paths and obstacles. The high-fidelity recognition results obtained from the 1-second-old frame are semantically robust; the system ensures that the identification of the environment is highly accurate, even if the spatial alignment requires fine-tuning.

To mitigate this 5% spatial offset while preserving the high-probability detection results, we implement **IMU-based predictive compensation**. While the server performs intensive inference over the 1-second window, the local ESP32-IMU continues to sample the head's rotation at 100Hz with near-zero latency. Before the final guidance is rendered, the Navigation Master fetches the cumulative IMU quaternion to calculate the total angle the head has turned during the 1000 ms t_{delay}. The system then mathematically re-projects the high-confidence YOLO segmentation mask by applying the inverse rotation to align the outdated visual data with the user's real-time pose. This sensor fusion strategy ensures that the directional guidance remains strictly accurate and stable within a 1.5% margin of the total FOV, effectively leveraging high-probability AI recognition through a low-latency mechanical bridge.

3. Cost

Component	Cost ($)	Quantity
Camera	150	1
AI Board (Jetson Nano/ ESP32)	300	1
Battery	50	1
Misc	100	1
GPU Leasing	200	1
IMU (ICM-42688-P)	120	1
Total	1020	

Table 2: Cost Breakdown

4. Schedule

The project schedule is divided into three primary phases: Component Evaluation and Baseline Development, System Integration and Prototype Assembly, and Performance Optimization and Real-World Testing. The following table details the weekly tasks assigned to each team member to ensure the successful completion of the Intelligent Wearable Vision System.

Week	Yi Su (ME)	Mingyan Gao (ECE)	Shengnan Cai (EE)	Junchen He (EE)
3/3	Hardware market research and component selection.	Define system architecture and communication protocols.	Research YOLO segmentation models for blind paths.	Investigate environmental perception algorithms.
3/10	Initial 3D modeling of the smart glasses frame.	Setup FastAPI server and WebSocket structure.	Prepare training datasets for obstacle detection.	Design street crossing logic and workflows.
3/17	Refine full-scale 3D modeling for sensor housing.	Develop master control orchestration logic.	Implement initial YOLO-based blind path segmentation.	Implement crosswalk and traffic light recognition.
3/24	Order hardware (ESP32-CAM, IMU, Battery).	Establish stable IO bridging and FastAPI communication.	Integrate YOLO models with the server backend.	Develop workflow for environmental perception.
3/31	**(Current)** Finish design blueprints for 3D printing.	**(Current)** Develop mode management in `app_main.py`.	**(Current)** Test initial obstacle detection reliability.	**(Current)** Optimize `workflow_cross_street.py`.
4/7	Assemble the prototype and hardware.	Optimize system latency and concurrent processing.	Conduct overall system integration testing.	Debug navigation workflows and coordinate data flow.
4/14	Integrate ESP32-CAM and IMU sensors into frame.	Refine state machine for dynamic feedback logic.	Improve detection reliability under motion.	Perform laboratory-based obstacle detection tests.
4/21	Finalize the compact wearable prototype assembly.	Final system debugging and latency benchmarking.	Fine-tune feedback design based on test results.	Conduct field tests in real outdoor scenarios.
4/28	Evaluate mechanical durability and safety.	Prepare data for the final report and analysis.	Verify accuracy against high-level requirements.	Finalize documentation and presentation slides.

Table 3: Weekly project schedule and individual task assignments.

5. Ethics and Safety

5.1 Ethical Considerations

As a wearable device designed to assist a vulnerable population (visually impaired individuals) in public spaces, this project presents several critical ethical considerations that must be addressed in alignment with the **IEEE Code of Ethics**. [1]

- **Data Privacy and Bystander Consent:** In accordance with IEEE Code of Ethics Tenet 1 (protecting the privacy of others), the continuous capture of video data in public spaces raises significant privacy concerns for bystanders. To address this, our system is designed with a strict *data-minimization* policy. The ESP32-CAM streams frames to the FastAPI server solely for real-time inference. No video data or images are permanently stored, recorded, or transmitted to third-party cloud services. Once a frame is processed for object detection, it is immediately discarded from the RAM.
- **Algorithmic Bias and Equitable Safety:** Machine learning models like YOLO can exhibit biases depending on their training data. If the model fails to detect certain demographics (e.g., children, individuals in wheelchairs) or struggles in specific neighborhoods (e.g., poorly lit areas), it could endanger the user. To mitigate this, we commit to using diverse, well-distributed datasets for training and validating our obstacle detection models, ensuring equitable performance across different scenarios.
- **System Transparency and Over-Reliance:** Users must not be misled about the system's capabilities. The device is an *assistive perception aid*, not a complete replacement for traditional mobility tools like white canes or guide dogs. We will clearly document the system's limitations (e.g., performance drops in heavy rain or extreme darkness) and instruct users to utilize the glasses as a supplementary tool.

5.2 Safety Considerations and Regulatory Standards

The physical and functional safety of the user is our highest priority. We have identified several potential hazards and implemented rigorous design decisions and mitigation procedures to address them.

- **Battery and Thermal Safety (UL 1642 Standard):** The system utilizes a 3.7V 1000mAh Lithium-ion battery located near the user's head. To prevent thermal runaway, fire, or chemical leakage, the battery circuit incorporates a dedicated Battery Management System (BMS) chip. This chip provides over-voltage, under-voltage, and short-circuit protection. The battery housing in the 3D-printed frame is structurally isolated and ventilated to ensure heat dissipation, complying with general UL 1642 safety guidelines for lithium batteries. [3]
- **Radio Frequency (RF) Exposure (FCC Part 15):** Because the ESP32 module continuously transmits data via Wi-Fi close to the human brain, RF exposure is a concern. We will ensure that the transmission power of the ESP32 is regulated and kept well within the Specific Absorption Rate (SAR) limits mandated by FCC Part 15 regulations for wearable consumer electronics.
- **Functional Fail-Safes and Danger Mitigation:** In the event of a system failure (e.g., loss of Wi-Fi connection to the server, server crash, or low battery), a sudden loss of guidance could leave the user

disoriented in a dangerous environment (like a crosswalk). To mitigate this, the Navigation Master includes a "heartbeat" monitoring mechanism. If the connection drops or latency exceeds 500ms, the system will immediately trigger a distinct, high-priority audio warning (e.g., "System Offline"), alerting the user to stop relying on the visual feedback and switch entirely to their white cane.

- **Auditory Safety and Environmental Awareness:** Visually impaired individuals rely heavily on ambient auditory cues (e.g., traffic sounds, footsteps) for spatial awareness. Delivering feedback via traditional earbuds would dangerously block these crucial sounds. Therefore, our design strictly dictates the use of open-ear audio feedback (such as bone-conduction transducers or directional micro-speakers integrated into the glasses frame) to ensure that the assistive feedback does not interfere with the user's natural hearing.

References

[1] *IEEE Code of Ethics*, IEEE Policies, Section 7.8, 2020. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: Apr. 8, 2026].

[2] Federal Communications Commission, *Evaluating Compliance with FCC Guidelines for Human Exposure to Radiofrequency Electromagnetic Fields*, OET Bulletin 65, Ed. 97-01, 1997.

[3] Underwriters Laboratories, *Standard for Lithium Batteries*, UL Standard 1642, 2012.

[4] G. Jocher *et al.*, "YOLO by Ultralytics," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics. [Accessed: Apr. 8, 2026].