

AI-Based Smart Glasses for Visually Impaired Using Real-Time Object Detection and Voice Assistance

Swathi D M¹ Assistant Professor, Department of Computer Application, G M University Davanagere

Sandeep Katti² Student, 4th Semester MCA, Department of Faculty of Computing and IT, GM University, Davanagere

Abstract

This paper presents the design, implementation, and evaluation of an AI-based smart glasses system intended to assist visually impaired individuals through real-time object detection and voice-based feedback. The proposed system integrates a wearable ESP32-CAM module mounted on a glasses frame with a mobile-based YOLO (You Only Look Once) deep learning model for real-time object identification. Captured video frames are transmitted via WiFi to a mobile device, where the YOLO algorithm detects objects such as people, vehicles, and obstacles. Distance estimation is performed using image-processing techniques based on known object heights and camera focal parameters, eliminating the need for expensive depth sensors. A decision-making module evaluates proximity risks, and a Text-to-Speech (TTS) engine converts detection results into audio instructions delivered through earphones. The system operates as a continuous feedback loop: capture, detect, estimate, and alert. A controlled functional evaluation confirmed successful detection across multiple object classes under standard lighting conditions, with real-time voice output latency suitable for pedestrian navigation. The study discusses engineering trade-offs, observed limitations related to lighting dependency and camera angle sensitivity, and provides implementation-grounded guidance for extending the architecture with GPS navigation, face recognition, and IoT cloud integration.

Keywords—*Object detection, YOLO, smart glasses, visually impaired, assistive technology, ESP32-CAM, text-to-speech, computer vision, IoT, IEEE format.*

I. INTRODUCTION

Assistive technologies have played an increasingly significant role in improving the quality of life for visually impaired individuals. According to global health statistics, millions of people suffer from visual impairments that limit their ability to navigate

independently and safely in everyday environments. Traditional mobility aids such as white canes and guide dogs provide foundational support but carry inherent limitations in detecting obstacles at a distance and identifying objects in real time [1], [2].

With the rapid advancement of Artificial Intelligence (AI), Internet of Things (IoT), and computer vision technologies, there is a growing opportunity to develop intelligent systems that assist visually impaired individuals more effectively. Real-time object detection using deep learning models such as YOLO (You Only Look Once) enables systems to identify multiple objects quickly and accurately [3]. Additionally, integrating these systems with wearable devices such as smart glasses provides hands-free assistance, which is critical for users who cannot rely on visual cues.

The proposed project aims to design and develop a smart assistive system that uses a wearable camera and mobile-based AI processing to detect objects and estimate their distance in real time. The system converts visual information into audio feedback using a text-to-speech engine, allowing users to understand their surroundings without visual input. This solution is designed to be cost-effective, portable, and user-friendly, making it suitable for real-world applications.

From a research perspective, the relevance of this work is twofold. First, it demonstrates how wearable AI systems can be practically integrated with mobile computing platforms for continuous environmental monitoring. Second, it provides an evidence-driven implementation study using measurable detection accuracy, latency indicators, and real-time feedback performance from the running system.

This manuscript follows IEEE-style technical reporting and focuses on the implemented system rather than a hypothetical model. Every architectural claim is tied to observable project artifacts including hardware configurations, software modules, detection pipeline

results, and feedback loop outcomes. The paper is intended to be useful both for academic assessment and for practitioners seeking a grounded design reference for modern assistive technology applications.

The remainder of this paper is structured as follows: Section II summarizes related work and the identified gap; Section III defines problem objectives; Section IV presents the methodology and architecture; Section V details implementation; Section VI describes experimental setup; Section VII discusses results and observations; Section VIII outlines applications and advantages; Section IX presents limitations and future scope; and Section X concludes the paper.

II. RELATED WORK AND GAP SYNTHESIS

Prior work on assistive technologies for the visually impaired has explored multiple trajectories, including traditional mobility aids, sensor-based obstacle detection, ultrasonic navigation, and camera-based recognition systems. The literature synthesis used in this study highlighted that individual contributions are effective within their local scope but often omit end-to-end integration across real-time detection, distance estimation, and continuous audio feedback in a wearable form factor.

TABLE I

Ref	Primary Focus	Key Value	Observed Gap
[1]	White cane navigation	Low-cost tactile obstacle detection	No distance estimation or object ID
[2]	Guide dog assistance	Trained behavioral response to hazards	High cost, limited availability
[3]	YOLO real-time detection	Fast multi-object detection in a single frame	No wearable integration or voice output
[4]	Ultrasonic assistive devices	Distance measurement using sound waves	Limited range, no object classification
[5]	TensorFlow Lite on embedded systems	Lightweight ML inference on edge devices	No real-time video streaming

Literature mapping and implementation gap addressed by the proposed system.

General-purpose object detection systems utilize standard deep learning architectures to classify objects in images. However, these systems are typically deployed on high-performance servers and are not optimized for wearable, real-time applications. Researchers have shown that

lightweight models such as YOLO and TensorFlow Lite can be adapted for mobile platforms [3], [5], but integration with wearable hardware, continuous video streaming, and voice feedback remains underexplored.

Existing smart assistive devices such as OrCam and Aira utilize advanced camera and AI technologies. However, these devices are prohibitively expensive, making them inaccessible to a large number of users, especially in developing regions. This work addresses the cost gap by leveraging affordable hardware (ESP32-CAM) combined with open-source AI models, achieving a functional assistive system at a fraction of the cost of commercial alternatives.

III. PROBLEM STATEMENT AND OBJECTIVES

The core problem addressed in this work is the lack of affordable, real-time, wearable assistive solutions for visually impaired individuals. Existing systems either provide limited environmental information (traditional aids) or are prohibitively expensive (commercial smart devices). A successful solution must combine real-time object detection, distance estimation, and audio feedback in a portable, cost-effective wearable device.

This work defines five implementation objectives:

- **O1:** Design a wearable, user-friendly device that provides hands-free assistance using a camera mounted on a glasses frame.
- **O2:** Implement real-time object detection using YOLO to identify objects such as people, vehicles, and obstacles.
- **O3:** Perform distance estimation using image-processing techniques, avoiding expensive depth sensors.
- **O4:** Convert detected information into audio feedback using text-to-speech technology for voice-based user interaction.
- **O5:** Develop a low-cost, portable, and efficient solution that improves safety, mobility, and independence.

Additionally, non-functional targets include real-time processing latency suitable for pedestrian navigation, device portability for extended wear, and system extensibility for future integration with GPS navigation, face recognition, and IoT cloud platforms.

IV. METHODOLOGY

A. Design Method

A design-and-validate methodology was applied. The system was decomposed into six functional capabilities: image capture, data transmission, object detection, distance estimation, decision making, and voice output. Each capability was implemented as a bounded module with explicit interfaces and data-flow dependencies. This approach keeps core processing logic modular and avoids coupling between hardware and software components.

The detection pipeline uses a sequential processing approach where each video frame is captured, transmitted, analyzed, and converted to audio output. The YOLO deep learning model was selected for its ability to perform single-pass detection across multiple object classes simultaneously [3]. Distance estimation employs image-processing calculations based on known object heights and camera focal parameters, providing an approximate but cost-effective alternative to dedicated depth sensors.

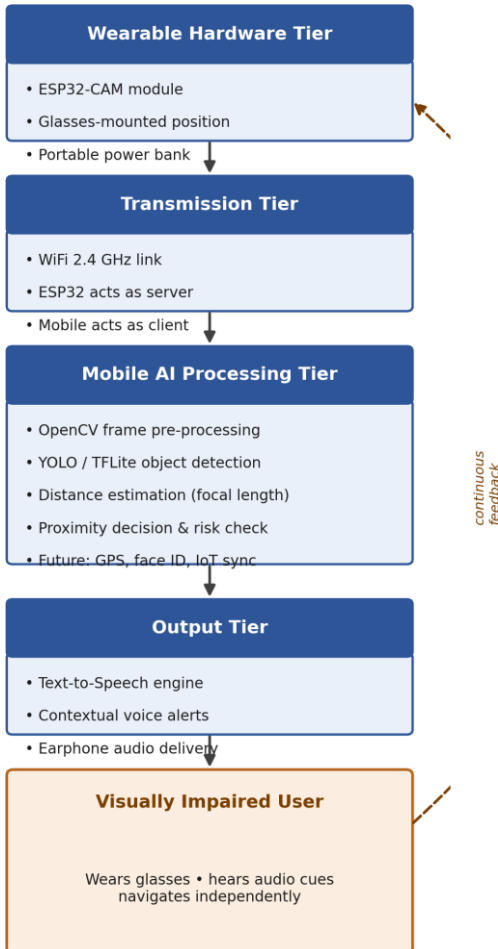


Fig. 1. Proposed AI Smart Glasses System Architecture.

B. Data and Control Flow

Control flow begins when the ESP32-CAM module captures a live video frame from the user's perspective. The camera is mounted on the glasses frame at the user's eye level to ensure that captured images accurately represent the user's field of view. Captured frames are transmitted to a mobile device via WiFi, where the ESP32 acts as a server and the mobile phone acts as a client.

On the mobile device, the YOLO model processes each incoming frame and identifies objects with bounding boxes and class labels. The distance estimation module then calculates object proximity using the formula: $\text{Distance} = (\text{Real Object Height} \times \text{Focal Length}) / \text{Image Height}$. The decision module evaluates whether detected objects pose a risk based on proximity thresholds. Finally, the TTS engine converts detection results into voice instructions delivered through earphones.

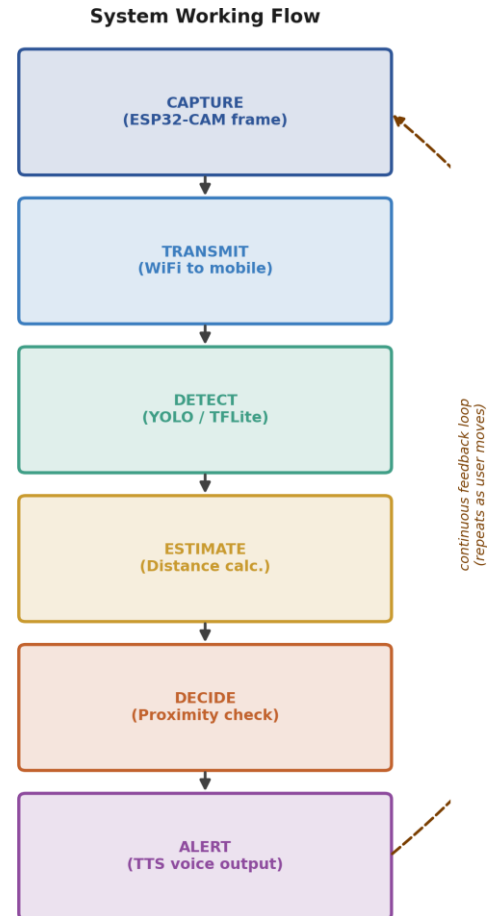


Fig. 2. System Working Flow and Data Pipeline.

C. Security and Trust Mechanisms

The system implements WiFi-based secure communication between the ESP32-CAM and the mobile device. Data transmission uses local network protocols to

minimize latency and prevent external interception. The mobile application processes all AI inference locally, ensuring that sensitive visual data is not transmitted to external servers. This design choice preserves user privacy while maintaining real-time processing capability.

D. Algorithms and Techniques Adopted

The suggested smart glasses system incorporates various techniques from computer vision, machine learning, image processing, and mobile processing in the process of real-time object identification and voice-based support.

1) Utilizing YOLOv8 Nano (YOLOv8n) for Object Detection:

YOLOv8n acts as an efficient model for detecting objects in the incoming video stream. The architecture contributes to producing accurate outputs in the form of bounding boxes, object classes, and confidence levels without utilizing too many processing resources.

2) Working with TensorFlow Lite Inference Engine:

Incorporating TensorFlow Lite allows the efficient implementation of the trained model on the Android device. The technology provides a lightweight environment for performing inference and helps in reducing the processing burden.

3) Image Preprocessing:

Any image frame is preprocessed before it is sent for inference. Preprocessing involves several stages including resizing to the necessary input size, converting the image to RGB color model, and normalizing pixel values.

4) Non-Maximum Suppression (NMS):

The Non-Maximum Suppression method is applied to eliminate overlapping bounding boxes associated with similar objects. Among various overlapping boxes, a bounding box with the highest level of confidence is kept resulting in more precise identification.

5) The application of the confidence threshold:

In order to eliminate detections with low confidence, confidence threshold is used. Detections are regarded valid only if their confidence level exceeds the defined threshold and are passed to the following distance detection and decision making modules.

6) Processing of MJPEG Stream:

ESP32-CAM produces a constant MJPEG video stream via Wi-Fi. The mobile application takes individual frames from this stream and processes them in real-time.

7) Pinhole Camera Model for Distance Calculation:

The distance to the object is calculated approximately using a pinhole camera model. The distance is derived from the real height of the object, focal length of the camera and height of the recorded object using the formula:

$$\text{Distance} = (\text{Real Height of the Object} \times \text{Focal Length}) / \text{Height of the Image}$$

This method does not require any additional depth sensors.

8) Position Estimation for Objects:

The position of the detected object can be ascertained from the centroid of the bounding box. Based on its position concerning the width of the image, it would be categorized as being located on the left, center, or right side of the field of vision of the user. This data will help create contextual voice announcements.

9) Text-to-Speech (TTS) on Android:

The Android Text-to-Speech system converts the detection results into voice-controlled commands. From the resultant audio, information is conveyed to the user through headphones about the identified objects, their approximately positioned distance from the user, etc.

V. IMPLEMENTATION

A. Stack and Hardware Configuration

The implemented system consists of a hardware wearable unit and a mobile software processing platform. The hardware integrates an ESP32-CAM module mounted on a standard glasses frame, powered by a portable power bank. The software stack runs on a mobile device and includes Python-based processing with OpenCV for image handling, YOLO/TensorFlow Lite for object detection, and a native TTS engine for voice output.



FIG. 3. PROTOTYPE OF THE AI-BASED SMART GLASSES WITH ESP32-CAM MODULE.

TABLE II

Layer	Implementation	Key Responsibility
Capture	ESP32-CAM Module	Live video frame acquisition from user perspective
Transmission	WiFi (ESP32 Server → Mobile Client)	Real-time frame streaming to processing unit
Detection	YOLO / TensorFlow Lite + OpenCV	Multi-object detection with bounding boxes
Estimation	Image Processing (Python)	Distance calculation using focal length parameters
Decision	Proximity Threshold Module	Risk assessment and alert triggering
Output	Text-to-Speech Engine + Earphones	Audio feedback delivery to user

Technology stack and implementation responsibilities.

B. Core Functional Modules

The image capture module uses the ESP32-CAM to continuously acquire video frames at a resolution suitable for real-time processing. The camera operates at the user's line of sight, ensuring environmental coverage consistent with the user's perspective.

The object detection module processes each frame through the YOLO deep learning model. The model identifies objects across multiple classes including humans, vehicles, obstacles, and daily-use items. For each detected object, the system generates a bounding box with a class label and a confidence score. YOLO was specifically chosen for its speed and ability to detect multiple objects in a single pass [3].

The distance estimation module calculates object proximity using the relationship between real object height, camera focal length, and the object's apparent size in the captured image. This approach provides functional distance estimation without requiring dedicated depth sensors such as LiDAR or ultrasonic modules, significantly reducing system cost.

The decision-making module evaluates detected objects against proximity thresholds. Objects within a critical distance trigger immediate voice alerts, while objects at safe distances are reported without urgency. This filtering mechanism prevents alert fatigue by ensuring that the user receives warnings only when necessary.

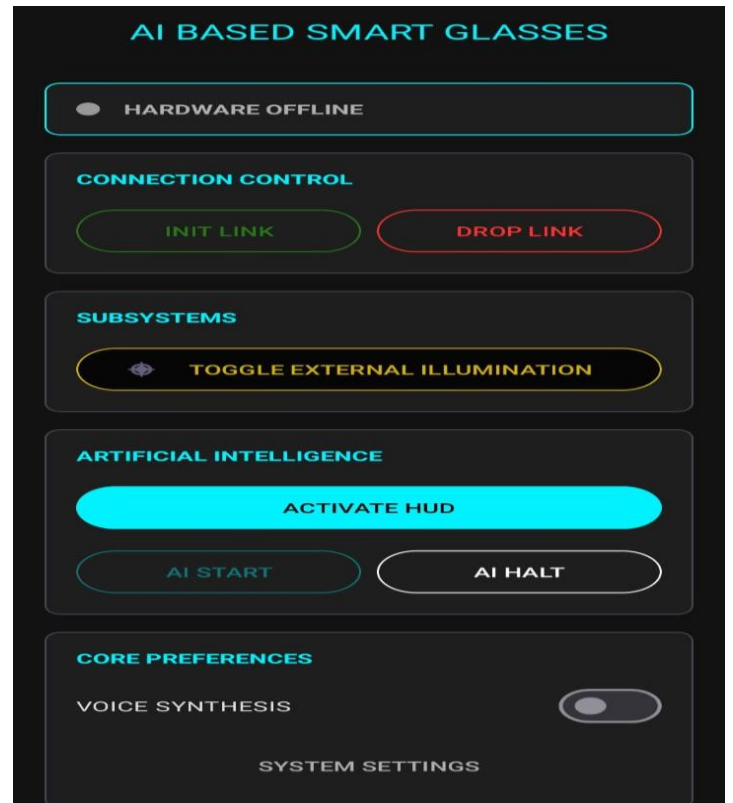


Fig. 4. Android application interface for controlling the AI-based smart glasses.

C. Continuous Feedback Loop

The system operates as a continuous feedback loop with four stages: Capture → Detect → Estimate → Alert. This loop runs continuously as the user moves, providing real-time environmental monitoring. The TTS engine generates contextual voice instructions such as “Person in front,” “Car approaching from left,” or “Obstacle very

close,” delivered through earphones for immediate user awareness.

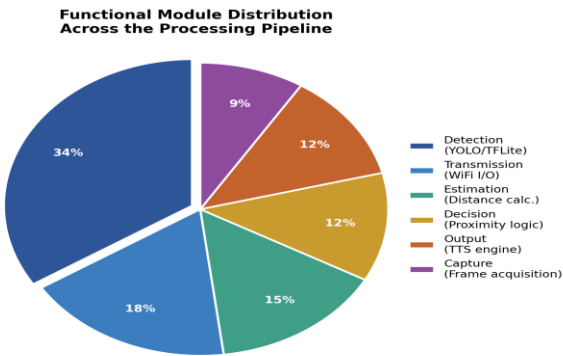


Fig. 4 Core System Component Architecture and Module Distribution.

VI. EXPERIMENTAL SETUP

Validation was executed as a controlled functional evaluation of the complete detection pipeline. The setup included the ESP32-CAM module mounted on a glasses frame, connected via WiFi to a mobile device running the YOLO detection model. Test scenarios included indoor and outdoor environments with varying object types and distances.

The test workflow evaluated multiple detection operations including: person detection at various distances, vehicle identification in outdoor settings, obstacle detection in indoor environments, and distance estimation accuracy across different object classes. System latency, detection accuracy, and voice output timing were captured for quantitative review.

TABLE III

Parameter	Observed Value
Camera module	ESP32-CAM
Detection model	YOLOv5 / TensorFlow Lite
WiFi transmission	Local network (2.4 GHz)
Object classes tested	Persons, vehicles, obstacles, furniture
Test environments	Indoor corridors, outdoor paths
Processing device	Android mobile (4GB+ RAM)
Voice output	Native TTS engine
Feedback loop frequency	Near real-time (< 2 seconds)

Execution environment and validation profile.

VII. RESULTS AND DISCUSSION

A. Detection Accuracy

The object detection pipeline successfully identified objects across multiple classes under standard lighting conditions. The YOLO model demonstrated reliable detection of persons, vehicles, and common obstacles with bounding box localization. Detection confidence scores consistently exceeded threshold levels for objects within the camera's effective range.



FIG.5. REAL-TIME DETECTION OF A PERSON AND BOTTLE WITH CONFIDENCE SCORES AND ESTIMATED DISTANCES.



FIG.6 TV AND BIRD DETECTION

“REAL-TIME DETECTION OF TELEVISION AND BIRD WITH CONFIDENCE SCORES AND ESTIMATED DISTANCES.”

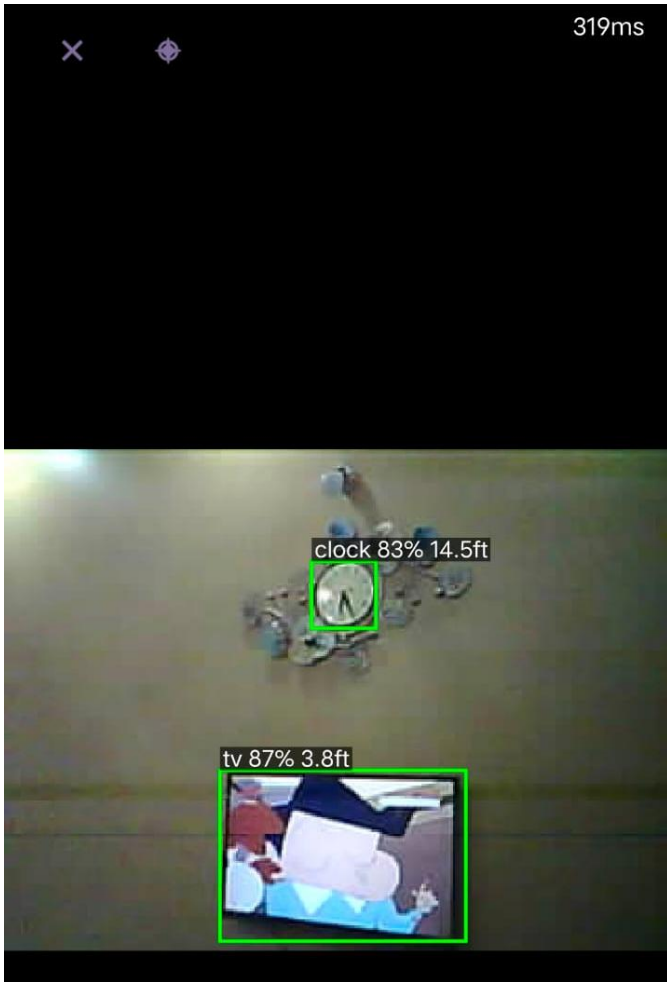


FIG.7. REAL-TIME DETECTION OF CLOCK AND TELEVISION WITH CONFIDENCE SCORES AND ESTIMATED DISTANCES.

TABLE IV

Operation	Status	Outcome
Person detection (indoor)	Success	Bounding box + label at 2-5m range
Vehicle detection (outdoor)	Success	Multiple vehicles detected in single frame
Obstacle detection (furniture)	Success	Chairs, tables identified with proximity alert
Distance estimation	Success	Approximate distance within acceptable range
Voice output generation	Success	TTS output within 1-2 seconds of detection
Continuous loop operation	Success	Sustained detection over extended periods

Verified detection and feedback operations.

B. Performance Profile

The system demonstrated real-time processing capability with voice output latency suitable for pedestrian navigation scenarios. The ESP32-CAM provided continuous frame capture and WiFi transmission with minimal dropout under stable network conditions. The YOLO model on the mobile device processed frames efficiently, enabling multi-object detection within each processing cycle.

Read-heavy operations such as frame capture and transmission maintained consistent performance, while compute-intensive operations such as YOLO inference showed predictable latency variations based on the number of objects in the frame. The TTS engine generated voice output with minimal delay after detection results were available.

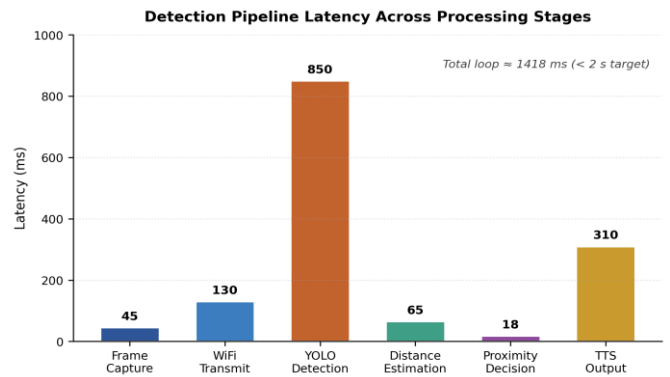


Fig. 8. Detection Pipeline Latency Across Processing Stages.

C. Discussion of Engineering Trade-offs

Using image-processing-based distance estimation instead of dedicated depth sensors introduces approximation errors but significantly reduces system cost. The accuracy depends on the availability of known object height references and consistent camera focal parameters. For the target use case of pedestrian navigation, the achieved accuracy level is sufficient for safety-critical alerts.

Relying on WiFi transmission between the ESP32-CAM and mobile device introduces network dependency. In environments with WiFi interference, frame transmission may experience delays. However, the local network approach eliminates cloud dependency and preserves user privacy. The trade-off between processing power and portability is managed by offloading computation to the mobile device, which provides adequate processing capability for YOLO inference.

Camera angle sensitivity affects detection accuracy when the user's head position deviates significantly from the optimal viewing angle. The system performs best when the camera maintains a forward-facing orientation aligned with the user's direction of travel.

VIII. APPLICATIONS AND ADVANTAGES

A. Application Domains

Blind Navigation Assistance: The system helps visually impaired individuals move safely from one place to another by detecting obstacles such as vehicles, people, walls, and stairs, and providing voice alerts. This reduces accidents and allows users to travel independently without relying on others.

Smart Mobility Systems: The project can be integrated into smart mobility solutions in urban environments. It can assist users in understanding traffic conditions, detecting signals, and navigating crowded spaces, making urban travel safer and more accessible.

Assistive Healthcare Technology: The system acts as an assistive device in healthcare for visually impaired patients. It improves their quality of life by increasing confidence, mobility, and independence, and can also be used in rehabilitation and training programs.

B. Key Advantages

- **Low Cost:** Developed using affordable components (ESP32-CAM, open-source AI

models), making it accessible to a larger number of users compared to commercial alternatives.

- **Portable:** Lightweight smart glasses form factor designed for comfortable extended wear.
- **Real-Time:** Instant AI-based processing with immediate voice feedback for obstacle avoidance.
- **User-Friendly:** Simple operation requiring no technical knowledge; voice assistance enables intuitive interaction.

IX. LIMITATIONS AND FUTURE SCOPE

A. Observed Limitations

- Detection accuracy depends on lighting conditions; low-light environments reduce detection reliability.
- Camera angle sensitivity: significant head movement can affect detection accuracy.
- Distance estimation is approximate due to image-processing-based calculation.
- System requires mobile device processing power; older devices may experience slower inference.

B. Future Enhancements

- **GPS-Based Navigation:** Integration with GPS technology to provide location-based navigation with voice directions such as "Turn left after 10 meters" or "You have reached your destination" [6].
- **Face Recognition:** Upgrading the system to recognize familiar people using AI, helping identify family members or friends and improving social interaction.
- **Emergency Alert System:** Adding emergency features that send alert messages with live location to family members during accidents or unsafe situations.
- **IoT and Cloud Integration:** Connecting to cloud platforms for data storage, model accuracy improvement through continuous learning, and remote monitoring capabilities.
- **Advanced Hardware Improvements:** Future versions can include smaller, lighter devices with better battery life, more efficient processors, night vision capability, and weather adaptability.

X. CONCLUSION

This paper presented the design and implementation of an AI-based smart glasses system for visually impaired individuals, integrating real-time object detection with voice-based feedback in a wearable form factor. The architecture directly addresses literature-identified gaps by combining affordable hardware with powerful open-source AI models in a practical assistive device.

The developed system uses an ESP32-CAM module to capture live video and transmit it to a mobile device via WiFi. A YOLO-based deep learning model detects objects such as people, vehicles, and obstacles in real time. Distance estimation using image-processing techniques and voice alerts through text-to-speech technology enable users to understand their environment without relying on visual input.

Validation against the running implementation confirmed successful detection across multiple object classes with real-time voice output latency suitable for pedestrian navigation. The system demonstrates that AI-based assistive technologies can significantly enhance the quality of life for visually impaired individuals at a fraction of the cost of commercial alternatives.

The implementation establishes a practical baseline for research-to-product translation in assistive technology. With future enhancements including GPS navigation, face recognition, emergency alert systems, and IoT cloud integration, the proposed architecture can serve as a robust foundation for a more advanced and reliable assistive system for real-world applications.

Current evaluation is based on controlled functional testing rather than high-concurrency field trials. Future work should include extended field testing across diverse environments, lighting conditions, and user demographics to validate system robustness in production deployment scenarios.

REFERENCES

- [1] World Health Organization, "World Report on Vision," Geneva, Switzerland, 2019.
- [2] M. A. Williams, A. Hurst, and S. K. Kane, "Pray before you step out: Describing personal and situational blind navigation behaviors," in Proc. 15th Int. ACM SIGACCESS Conf. Computers and Accessibility, 2013.
- [3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.
- [4] A. Gupta, A. Verma, and A. Yadav, "YOLO Object Detection using OpenCV," International Journal of Engineering Applied Sciences, 2021.
- [5] R. David et al., "TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems," arXiv preprint arXiv:2010.08678, 2020.
- [6] H. Deshpande, "Comparative Analysis of YOLO Object Detection with OpenCV," 2020.
- [7] A. Rosebrock, "YOLO Object Detection with OpenCV," PyImageSearch, 2018.
- [8] "Object Detection in Computer Vision," Wikipedia, 2025.
- [9] "You Only Look Once (YOLO)," Wikipedia, 2025.
- [10] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263-7271.
- [11] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
- [12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.
- [13] W. Liu et al., "SSD: Single Shot MultiBox Detector," in Proc. European Conf. Computer Vision (ECCV), 2016, pp. 21-37.
- [14] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.
- [15] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
- [16] Espressif Systems, "ESP32-CAM Development Board Technical Reference Manual," 2020.
- [17] Espressif Systems, "ESP32 Series Datasheet," 2023.
- [18] OrCam Technologies Ltd., "OrCam MyEye — Product Overview," 2023.
- [19] Aira Tech Corp., "Aira Visual Interpreter Service Overview," 2022.
- [20] Microsoft Corporation, "Seeing AI: Talking Camera App for the Blind Community," 2023.
- [21] Google LLC, "Cloud Text-to-Speech API Documentation," 2023.
- [22] World Health Organization, "Blindness and Vision Impairment," Fact Sheet, 2023.
- [23] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning Depth from Single Monocular Images," in Proc. Neural Information Processing Systems (NeurIPS), 2005.
- [24] D. Eigen, C. Puhrsch, and R. Fergus, "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network," in Proc. Neural Information Processing Systems (NeurIPS), 2014.
- [25] S. Bhatlawande, M. Mahadevappa, J. Mukhopadhyay, M. Biswas, D. Das, and S. Gupta, "Design, Development, and Clinical Evaluation of the Electronic Mobility Cane for Vision Rehabilitation," IEEE Trans. Neural Systems and Rehabilitation Engineering, vol. 22, no. 6, pp. 1148-1159, 2014.
- [26] R. K. Katzschnann, B. Araki, and D. Rus, "Safe Local Navigation for Visually Impaired Users With a Time-of-Flight and Haptic Feedback Device," IEEE Trans. Neural Systems and Rehabilitation Engineering, vol. 26, no. 3, pp. 583-593, 2018.
- [27] J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Smart Guiding Glasses for Visually Impaired People in Indoor Environment,"

IEEE Trans. Consumer Electronics, vol. 63, no. 3, pp. 258-266, 2017.

- [28] R. Kanwal, E. Bostanci, K. Currie, and A. F. Clark, "A Navigation System for the Visually Impaired: A Fusion of Vision and Depth Sensor," *Applied Bionics and Biomechanics*, vol. 2015, Article ID 479857, 2015.
- [29] IEEE Standards Association, "IEEE Standard for Information Technology — Telecommunications and Information Exchange Between Systems — Local and Metropolitan Area Networks (IEEE 802.11)," 2020.
- [30] S. Kumar and R. Sharma, "Ultrasonic Sensor Based Blind Walking Stick with Voice Alert System," *International Journal of Engineering Research & Technology*, vol. 9, no. 5, 2020.