Indian Journal of Engineering Research Networking and Development
Volume: 3 Issue: 03 | March 2026 | www.ijernd.com

Voice Assistant for Blind People: An AI-Driven Mobile System Using YOLOv8 and MiDaS for Real-Time Object Detection and Depth Estimation

G. Vijaya Lakshmi 1,B. Ganesh 2, G. Viswateja 3 ,T.sony Swaroop 4

1 Assistant professor, Dept Of Computer Science And Engineering, Sanketika Institute Of Technology And Management, Visakhapatnam, Andhra Pradesh, India

2 Student, Dept of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India

3Student, Dept of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India

4Student, Dept of Computer Science and Engineering, Sanketika Institute of Technology and Management, Visakhapatnam, Andhra Pradesh, India

Abstract

Visual impairment presents profound barriers to independent navigation, creating a persistent demand for intelligent assistive technologies. This paper presents a mobile-based Voice Assistant system that empowers visually impaired individuals by transforming their physical surroundings into real-time auditory descriptions. The proposed architecture couples a Flutter cross-platform mobile frontend with a Python-based backend server, employing YOLOv8 for high-speed object detection and MiDaS for monocular depth estimation. A captured image is transmitted via RESTful API to a FastAPI server, where objects are identified and their approximate distances are derived by correlating bounding-box centroids with the corresponding depth map regions. The resulting structured data is converted to natural speech through on-device Text-to-Speech (TTS), delivering descriptive alerts such as "person detected at 1.5 metres." MongoDB Atlas handles user authentication and data persistence, while the client-server design keeps the mobile application lightweight. Experimental trials across indoor and outdoor environments demonstrate a mean object-detection precision of 87.4% and an average response latency of 1.3 seconds. This integrated, hardware-agnostic solution provides a cost-effective, portable, and scalable approach to assistive technology, significantly improving situational awareness and independence for blind users.

***Index Terms*—** voice assistant, visual impairment, YOLOv8, MiDaS, depth estimation, assistive technology, object detection

I. Introduction

Visual impairment is a significant global challenge affecting an estimated 2.2 billion people worldwide, according to the World Health Organization [1]. Among these, over 36 million individuals are classified as blind, and a large proportion lack access to sophisticated navigational aids. Traditional tools such as white canes and guide dogs provide basic mobility support; however, they offer limited capacity to identify specific objects or estimate distances, constraining the user's environmental awareness.

The convergence of deep learning, mobile computing, and cloud-based APIs has created new opportunities for developing intelligent assistive systems that are both powerful and affordable. Smartphones, equipped with high-resolution cameras

and processing capabilities, serve as versatile platforms for deploying computer vision pipelines without the cost of specialized hardware.

The proposed *Voice Assistant for Blind People* leverages these technological advances to address the shortcomings of existing solutions. The system captures ambient images through a standard smartphone camera, processes them on a cloud-connected backend using state-of-the-art models—YOLOv8 [2] for object detection and MiDaS [3] for monocular depth estimation—and delivers auditory scene descriptions via Text-to-Speech (TTS). The mobile frontend is developed using Flutter [4], ensuring cross-platform compatibility, while FastAPI [5] powers the backend. User data and authentication are managed through MongoDB Atlas.

The primary contributions of this work are: (i) an end-to-end, hardware-agnostic assistive pipeline combining real-time object detection and depth estimation; (ii) a lightweight, accessible Flutter application with integrated TTS; and (iii) empirical evaluation demonstrating practical accuracy and latency suitable for real-world deployment.

II. Related Work

Research in assistive technology for the visually impaired spans multiple disciplines. Early systems relied on ultrasonic range sensors [6] and radio-frequency identification (RFID) tags to detect obstacles and provide haptic or audio feedback. While effective for simple obstacle avoidance, these approaches lack the semantic richness needed for comprehensive scene understanding.

The emergence of deep convolutional neural networks transformed object detection. Redmon et al. introduced YOLO [2], a single-pass architecture that performs detection in real time, subsequently refined through YOLOv3 [7], YOLOv4 [8], and YOLOv8 [9]. These models have been applied to assistive wearables by Bai et al. [10], who demonstrated an accuracy of 84% for indoor obstacle detection on a Raspberry Pi-based system.

Depth estimation approaches range from stereo-camera systems [11] to monocular methods. Ranftl et al. introduced MiDaS [3], achieving robust zero-shot cross-dataset generalization through training on diverse data sources. Compared with FCRN [12] and DenseDepth [13], MiDaS provides superior performance on unconstrained images, making it well suited for mobile assistive applications.

On the application side, Microsoft Seeing AI [14] combines OCR, object recognition, and scene description. While comprehensive, it relies on persistent cloud inference and lacks configurable distance feedback. Be My Eyes [15] employs crowdsourced sighted volunteers, introducing latency and unavailability issues. NavCog [16] provides indoor navigation via Bluetooth beacons but requires environment-specific infrastructure. In contrast, the proposed system integrates detection, depth, and speech into a self-contained, infrastructure-free solution.

III. Methodology / System Design

A. System Architecture

The proposed system follows a client-server architecture comprising three principal tiers: a Flutter mobile client, a Python-FastAPI backend, and a MongoDB Atlas database. The overall pipeline is illustrated in Fig. 1.

Flutter AppCameraCaptureHTTP POSTTTS OutputFastAPI BackendYOLOv8Object DetectionMiDaSDepth EstimationNMS + FusionResult JSONImage Pre-processingOpenCV / NumPyMongoDBAtlasUser AuthData StorageJSON Response→ TTS SpeechClient–Server Architecture

Fig. 1. System architecture block diagram illustrating the client-server pipeline.

B. Object Detection Module (YOLOv8)

Object detection is executed on the backend using YOLOv8 [9], a single-stage detector that processes an image in one forward pass. The pre-trained YOLOv8n (nano) variant is used for its balance of accuracy and inference speed. Given an input image I of dimensions $H \times W \times 3$, the network predicts a set of bounding boxes $\{b_i\}$ along with class labels $\{c_i\}$ and confidence scores $\{p_i\}$. Non-Maximum Suppression (NMS) is applied to filter redundant predictions:

$$b^* = NMS(\{b_i\}, threshold = 0.45) \quad (1)$$

Only detections with $p_i \geq 0.5$ are forwarded to the depth module, reducing computational load and false positives.

C. Depth Estimation Module (MiDaS)

Monocular depth estimation is performed using the MiDaS DPT-Hybrid model [3]. The model outputs a relative inverse depth map $D(x,y)$ for each pixel. To derive a normalized distance score d_i for a detected object with bounding box b_i, the mean depth within the box region is computed and inverted:

$$d_i = 1 / mean(D[y_1{:}y_2, x_1{:}x_2]) \quad (2)$$

A calibration mapping f: $d_i \rightarrow$ metres is applied using a linear regression fitted on a set of reference measurements at known distances, yielding approximate metric distances suitable for auditory narration.

D. Flutter Mobile Application

The frontend is built with Flutter, enabling deployment on both Android and iOS from a single Dart codebase. The application consists of three primary screens: Authentication (Login/Signup), Home (Image Capture), and Result (TTS Playback). The image capture flow utilises the `image_picker` package, and the `http` package dispatches multipart POST requests to the FastAPI endpoint. Voice output is rendered via `flutter_tts`, converting the JSON-encoded detection results into natural language sentences.

User Login/SignupCameraCaptureHTTP POSTto FastAPIJSONResponseTTS AudioOutput to UserFlutter Application Flow

Fig. 2. Flutter application flow from login to voice output.

E. Text-to-Speech Integration

The TTS module receives a structured list of detected objects and their distances from the JSON response. A natural-language formatter constructs sentences of the form *"[Object] detected at approximately [distance] metres."* The `flutter_tts` package converts these strings to audio. Parameters including speech rate (0.5×), pitch (1.0), and volume (1.0) are adjustable by the user for personalised accessibility.

F. Backend API Design

The FastAPI server exposes a single inference endpoint `POST /detect` that accepts a multipart image upload. Upon receipt, the image is decoded with OpenCV, resized to 640×640 for YOLOv8, and concurrently passed to MiDaS for depth map generation. Detected bounding boxes are matched with depth values via equation (2), and the aggregated results are serialised as JSON. MongoDB Atlas is queried via PyMongo for token-based user authentication on all routes.

IV. Results & Discussion

A. Experimental Setup

The system was evaluated on a dataset of 500 test images captured across four environments: indoor corridors, outdoor pathways, retail spaces, and home interiors. Ground-truth bounding boxes were annotated manually. Object detection performance was assessed using standard metrics: Precision, Recall, and mean Average Precision (mAP@0.5). Depth accuracy was evaluated by comparing normalised predicted distances to LiDAR ground-truth measurements.

TABLE I

Object Detection Performance by Environment

Environment	Precision (%)	Recall (%)	mAP@0.5 (%)
Indoor Corridor	89.2	86.7	88.1
Outdoor Pathway	85.4	83.1	84.3
Retail Space	88.6	85.9	87.2
Home Interior	90.1	88.4	89.8
Overall	**88.3**	**86.0**	**87.4**

TABLE II

System Latency Breakdown (ms)

Component	Mean (ms)	Std Dev (ms)
Image Upload	310	45
YOLOv8 Inference	420	38

MiDaS Inference	510	52
JSON Serialisation	30	8
TTS Playback (start)	60	12
Total E2E	**1330**	**155**

B. Detection Performance

As reported in Table I, the system achieves an overall mAP@0.5 of 87.4% across all test environments. Performance is marginally lower in outdoor scenarios due to variable lighting and dynamic backgrounds, consistent with known limitations of CNN-based detectors [9]. Indoor environments yield the best results owing to controlled illumination and stable visual features.

02040608010088.184.387.289.8IndoorOutdoorRetail HomemAP@0.5 (%)

Fig. 3. mAP@0.5 performance across four evaluation environments.

C. System Latency

Table II presents the end-to-end latency decomposition. The total mean latency of 1,330 ms (1.3 s) is primarily dominated by the dual inference stage (YOLOv8 + MiDaS, ≈ 930 ms combined). This is acceptable for a real-world assistive scenario, where periodic snapshot-based feedback—as opposed to continuous video streaming—is the primary interaction mode. Network transmission accounts for 310 ms on average over a 4G LTE connection. Deployment on GPU-enabled servers is expected to reduce inference time to below 200 ms.

D. Comparison with Related Systems

TABLE III

Comparison with Existing Assistive Systems

System	Detection	Depth	TTS	HW-Free
Seeing AI [14]	✓	✗	✓	✓
Be My Eyes [15]	Human	✗	✗	✓
Bai et al. [10]	✓	✗	✓	✗
NavCog [16]	✗	✗	✓	✗
Proposed	✓	✓	✓	✓

As shown in Table III, the proposed system is the only solution that simultaneously offers object detection, monocular depth estimation, TTS output, and hardware-free operation. This positions it as a comprehensive yet accessible alternative to existing tools.

V. Conclusion & Future Work

This paper has presented a mobile-based Voice Assistant for visually impaired individuals that integrates YOLOv8 object detection, MiDaS depth estimation, and TTS feedback within a Flutter-FastAPI-MongoDB architecture. The system achieves a mean mAP@0.5 of 87.4% and an end-to-end latency of 1.3 seconds on commodity hardware, demonstrating practical viability for real-world deployment. By operating entirely on standard smartphones without specialised hardware, the solution is accessible, portable, and cost-effective.

Future work will focus on ten key enhancements: (1) real-time video stream processing using frame-by-frame detection; (2) offline inference via on-device quantised models; (3) GPS-integrated outdoor navigation with turn-by-turn guidance; (4) LiDAR-calibrated metric distance conversion; (5) multi-language TTS support; (6) personalised voice profiles; (7) OCR-based text reading for signs and documents; (8) GPU-accelerated cloud deployment for sub-200 ms inference; (9) smart-glass and wearable device integration with haptic feedback; and (10) enhanced privacy controls and end-to-end encryption. These advances aim to evolve the system into a comprehensive, robust assistive platform that maximises independence and quality of life for blind users.

References

[1] World Health Organization, *World Report on Vision*, WHO Press, 2019.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE CVPR*, 2016, pp. 779–788.

[3] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1623–1637, 2022.

[4] Google LLC, "Flutter – Build apps for any screen," Flutter Documentation, 2023. [Online]. Available: https://flutter.dev/docs

[5] S. Ramírez, "FastAPI – Modern, fast web framework for building APIs with Python," FastAPI Documentation, 2023. [Online]. Available: https://fastapi.tiangolo.com

[6] G. Maidenbaum, S. Abboud, and A. Amedi, "Sensory substitution: Closing the gap between basic research and widespread practical visual rehabilitation," *Neurosci. Biobehav. Rev.*, vol. 41, pp. 3–15, 2014.

[7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[9] Ultralytics, "YOLOv8 Object Detection Model," Ultralytics Documentation, 2023. [Online]. Available: https://docs.ultralytics.com

[10] J. Bai, Z. Liu, Z. Lin, Y. Li, S. Lian, and D. Liu, "Wearable travel aid for environment perception and navigation of visually impaired people," *Electronics*, vol. 8, no. 6, p. 697, 2019.

[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE CVPR*, 2012, pp. 3354–3361.

[12] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. IEEE 3DV*, 2016, pp. 239–248.

[13] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," *arXiv preprint arXiv:1812.11941*, 2018.

[14] Microsoft Corporation, "Seeing AI – Talking camera app for the blind and low vision community," Microsoft Research, 2023. [Online]. Available: https://www.microsoft.com/en-us/ai/seeing-ai

[15] Be My Eyes, "Be My Eyes – Helping blind and low-vision people lead more independent lives," 2023. [Online]. Available: https://www.bemyeyes.com

[16] K. Ahmetovic, C. Gleason, C. Ruan, K. Kitani, H. Takagi, and C. Asakawa, "NavCog: A navigational cognitive assistant for the blind," in *Proc. ACM MobileHCI*, 2016, pp. 90–99.