



International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Object Detection System for Visually Impaired People Using Yolo Technology

K. Sujitha, Dr J Evangelin Deva Sheela

Department of Computer Application, Sarah Tucker College, Tirunelveli

ABSTRACT

Visually impaired individuals often face challenges in perceiving nearby objects and maintaining awareness of their surrounding environment. This paper presents a low-cost, real-time object detection assistant that uses a webcam, computer vision, and speech output to provide environmental awareness. Live camera frames are processed using OpenCV and analysed using the YOLO-World object detection model. The system detects selected everyday obstacles, including persons, chairs, doors, vehicles, walls, and stairs. A priority mechanism ranks objects according to their position in the camera frame and relative bounding-box size, giving greater importance to objects located ahead of the user. Relative object size is used to provide an apparent proximity category: very close, near, or far. The most relevant obstacle is converted into a spoken alert using Windows Text-to-Speech, while a Streamlit dashboard displays the live frame, object labels, confidence scores, positions, and priority status. The work is designed as an accessible prototype that can run on a standard computer with a webcam and speaker. A structured evaluation protocol is included to measure detection performance, response time, and behaviour under different lighting conditions before journal submission.

Keywords: visual impairment, assistive technology, object detection, YOLO-World, OpenCV, text-to-speech, real-time computer vision

1.Introduction

The ability to identify and understand objects in the surrounding environment is important for the independent living and mobility of visually impaired people. Individuals with visual impairment may face difficulties in recognizing common objects, understanding their immediate surroundings, and responding to environmental situations without assistance. Everyday objects such as chairs, bottles, people, vehicles, bags, and other obstacles may be difficult to identify when visual information is unavailable. Traditional assistive methods such as walking canes, guide dogs, and human assistance provide valuable support, but they may not always provide information about the type of objects present in the surrounding environment. Therefore, intelligent assistive technologies are increasingly being explored to provide additional environmental awareness [1], [4].

The development of **Artificial Intelligence (AI), Machine Learning, Deep Learning, and Computer Vision** has created new possibilities for developing assistive systems. Computer vision enables computers to analyse images and video, while deep learning models can learn visual patterns and recognize objects automatically. Object detection is particularly useful because it performs both **object classification and localization**, allowing a system to determine what an object is and where it appears within an image or video frame. With lightweight deep learning models and affordable computing devices, real-time object detection can be implemented using ordinary computers and webcams.

Among the available object detection techniques, the **YOLO (You Only Look Once)** family has gained significant attention because of its ability to perform object detection efficiently in real time [2]. Unlike image classification, which generally identifies the main class of an image, object detection can identify multiple objects in the same frame using bounding boxes and class predictions. YOLO processes the image through a neural network and produces object detection results efficiently, making it suitable for applications such as surveillance, robotics, autonomous systems, and assistive technologies. In the proposed system, **YOLO-World** is used for real-time object detection and supports an open-vocabulary approach, allowing the system to work with a selected set of relevant everyday objects [3], [8].

The proposed **AI-Based Object Detection System for Visually Impaired People Using YOLO Technology** is designed to detect objects from a live webcam feed and provide useful information through audio and visual feedback. OpenCV is used to capture and process the live video frames, while YOLO-World analyses the frames and identifies selected objects such as persons, chairs, doors, vehicles, walls, stairs, and other obstacles. Each detection provides an object class, confidence score, and bounding-box coordinates. The system further analyses the position and relative size of detected objects to classify them as **left, ahead, or right** and to estimate their apparent proximity as **very close, near, or far**. A priority mechanism is used to select the most relevant object for the current scene [5], [6].

Voice feedback is an important part of the proposed system because visually impaired users may not be able to depend on visual output. The selected object information is converted into speech using a Text-to-Speech mechanism. The generated alert can include the object name, position, and apparent proximity, providing the user with concise information about the surrounding environment. A repeated-alert control mechanism helps prevent the same

object from being announced continuously. A **Streamlit dashboard** is also provided to display the live video, detected objects, bounding boxes, confidence scores, positions, and priority information for monitoring and evaluation.

The proposed system is developed as a low-cost and accessible prototype that can operate using a standard computer, webcam, and speaker or headphones. The current implementation uses a pretrained YOLO-World model and does not involve training on a custom visually impaired dataset. In addition, apparent proximity is based on the relative size of the detected bounding box and does not represent the actual physical distance of an object. Future improvements can include depth-based distance measurement, object tracking, custom dataset training, OCR, navigation assistance, multilingual speech, mobile integration, and wearable or smart-glasses deployment. Overall, the project demonstrates how AI, computer vision, and speech technology can be combined to provide an accessible approach to environmental awareness for visually impaired users.

2. Related Work

Recent work in assistive computer vision has combined camera-based object detection with speech output [4], mobile direction guidance [5], object tracking [6], and dual-camera evaluation [7]. Table 1 summarises selected studies relevant to the proposed system.

Study	Approach	Main contribution / limitation
Najm et al. (2024) [4]	YOLO, OpenCV and Google Text-to-Speech	Detected objects from webcam video and announced object location through speech; focused on general audio feedback.
Cheng et al. (2024) [3]	YOLO-World	Introduced real-time open-vocabulary detection using text prompts; supports detection beyond a fixed label set.
Bamane et al. (2024) [5]	Android, YOLO, OpenCV and sensor data	Combined object detection with direction prediction and audio/haptic guidance for mobile use.
Joshna et al. (2025) [6]	SSD MobileNet V3, OpenCV and pyttsx3	Implemented webcam-based detection, tracking, bounding boxes and voice alerts on resource-limited devices.
Nabila et al. (2025) [7]	YOLOv10, dual camera and speaker	Tested a dual-camera assistive system in different lighting and distance conditions; low light remained challenging.

Table 1. Summary of related assistive object-detection studies.

The reviewed work shows the usefulness of real-time detection and speech output for visually impaired users [4-7]. However, practical systems must avoid information overload and should prioritise the object most relevant to immediate movement. The proposed system addresses this requirement by ranking detected objects based on their path relevance and apparent proximity.

3. Problem Statement

The problem addressed in this work is the lack of immediate, understandable information about nearby obstacles for a visually impaired user. A user may need to know not only the object name, but also whether the object is directly ahead and whether it appears close. The objective is to provide this information in real time through a low-cost software prototype.

4. Objectives

To capture and process live video from a standard webcam.

To detect common surrounding objects and obstacles using YOLO-World.

To identify the approximate position of an obstacle as ahead, left, or right.

To prioritise an object in the walking path and estimate apparent proximity from its relative image size.

To provide repeated, understandable voice alerts and a live visual dashboard.

5. Proposed System

The proposed system is implemented in Python. A camera supplies live frames to OpenCV, which reads and prepares the image for inference. YOLO-World is configured with a custom vocabulary of daily obstacles. For every detection, the system obtains the object label, confidence score, and bounding-box coordinates. These coordinates are used to determine the object's position and relative size in the frame.

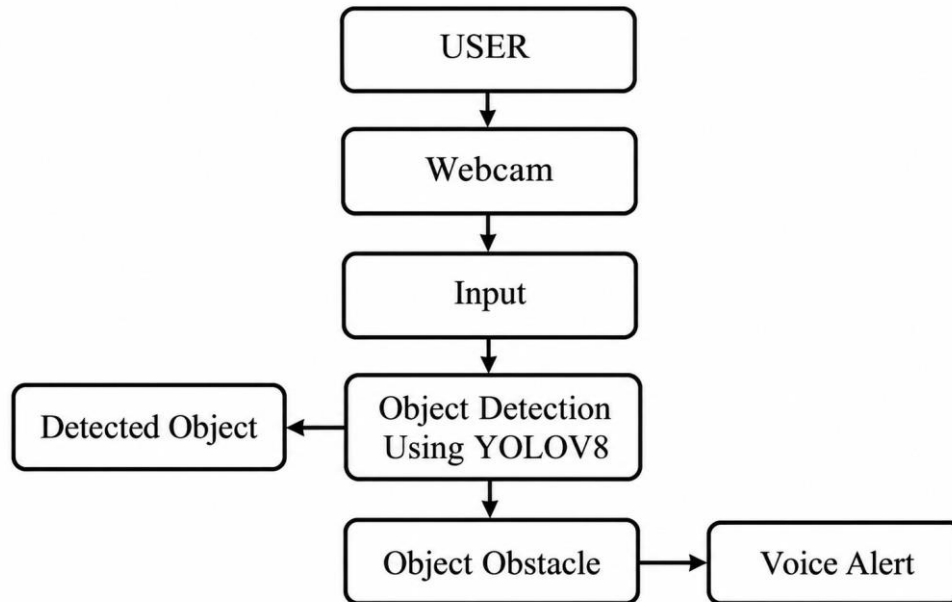


Figure 1. Architecture of the proposed AI-based obstacle detection assistant.

5.1 System Modules

Module	Function
Live Video Capture	Captures the surrounding environment continuously through a laptop webcam.
Frame Extraction	Separates the live webcam video into individual frames for continuous processing.
Frame Processing	OpenCV reads the newest frame, controls resolution and performs basic image preparation.
Object Detection Using YOLO-World	YOLO-World detects objects and returns labels, confidence scores and bounding boxes.
Object Recognition and Priority Selection	Determines the object position and apparent proximity, then selects the most relevant obstacle.
Voice Alert	Windows Text-to-Speech converts the selected object alert into audible guidance.
Live Dashboard and Display Output	Streamlit displays the annotated live frame, object details, priority status and apparent proximity.

Table 2. Functional modules of the proposed system.

6. METHODOLOGY

6.1 Video Input

The system begins by capturing real-time video using a webcam connected to the computer. The webcam continuously observes the user's surrounding environment and provides a live video stream as the primary input to the system. This allows the system to identify objects dynamically without requiring the user to manually capture or upload images.

The camera input is accessed using OpenCV. The video stream is continuously monitored so that newly appearing objects can be detected. Real-time visual input is important for assistive systems because the surrounding environment may change frequently while the user is moving [1], [4].

6.2 Frame Extraction

The continuous webcam video is divided into individual image frames using OpenCV. Each frame represents the visual information captured by the camera at a particular moment.

The extracted frames are passed through the detection pipeline one after another. Processing individual frames allows the YOLO-World model to analyse the current surroundings and identify objects present in the scene. Continuous frame extraction also enables the system to respond when objects appear, disappear, or change their position.

The system focuses on processing the latest available frame to maintain responsive operation and reduce unnecessary delay.

6.3 Frame Processing

After extraction, the captured frame is prepared for object detection. OpenCV manages the camera input and provides the frame to the YOLO-World detection model.

The frame contains the visual information required for identifying objects such as people, vehicles, chairs, doors, stairs, poles, and other objects included in the system vocabulary. The processed frame is then passed to the detection stage for further analysis.

The frame-processing stage acts as a connection between the webcam and the AI model, ensuring that the live camera information can be continuously supplied to the object detection system.

6.4 Object Detection

The YOLO-World model is used to detect objects present in each processed video frame. YOLO is a real-time object detection approach that performs object localization and classification efficiently [2]. YOLO-World extends this concept by supporting open-vocabulary object detection, allowing objects to be detected using a specified vocabulary [3], [8].

The proposed system uses an obstacle-oriented vocabulary containing objects such as person, bicycle, motorcycle, car, bus, truck, chair, table, sofa, bed, door, wall, stairs, step, curb, pole, pillar, tree, bench, traffic light, traffic sign, animal, and bag.

For every detected object, the model provides the **object class, confidence score, and bounding-box coordinates**. A confidence threshold is applied so that detections with insufficient confidence can be ignored. This helps reduce unreliable detections and unnecessary voice alerts.

6.5 Object Recognition & Priority Selection

After object detection, the system identifies the class of each detected object and analyses its location within the camera frame. The centre of the bounding box is used to determine whether the object is positioned on the **left, ahead, or right** of the user.

When several objects are detected simultaneously, the system does not announce every object at the same time. Instead, it analyses the detected objects and selects the most important object based on its position and relative size. Objects appearing in the central region and occupying a larger portion of the frame can receive higher priority.

The system also provides an **apparent proximity** classification. The relative size of the bounding box is used to classify an object as **very close, near, or far**. This is only an image-based approximation and does not represent the actual physical distance of the object in metres. Direction and priority-based object assistance can provide useful additional information for visually impaired users [5], [6].

6.6 Voice Alert

The selected high-priority object is converted into an audio message using a **Text-to-Speech (TTS)** service. Instead of depending only on visual information, the user receives information through spoken alerts.

The generated message can include the object's name, relative position, and apparent proximity. For example, the system may provide alerts such as “Person very close,” “Chair on your left,” or “Car ahead.”

A repeated-alert control mechanism is used to prevent the same information from being announced continuously while the object remains visible in consecutive frames. This helps make the audio output clearer and reduces unnecessary speech. Voice-based object feedback has been explored as an important component of assistive systems for visually impaired users [4], [7].

6.7 Display Output

The final detection results are displayed through the **Streamlit dashboard**. The dashboard provides a visual representation of the processed video and the objects detected by the system.

Detected objects are displayed using bounding boxes and labels. The dashboard can also present information such as the detected object, its position, apparent proximity, priority status, and current voice alert.

Although the visual dashboard is mainly useful for system monitoring, testing, and demonstration, the voice output provides the primary accessible information for the intended visually impaired user.

7. Implementation

Component	Technology used
Programming language	Python
Object detection	Ultralytics YOLO-World model (yolov8s-worldv2.pt)
Video and image processing	OpenCV
Speech output	Windows SpeechSynthesizer / Text-to-Speech
Dashboard	Streamlit with automatic refresh
Hardware	Laptop/desktop, webcam and speaker

Table 3. Software and hardware components.

The full project version provides voice alerts and a dashboard. A smaller Modules 1-3 version is also retained for demonstration of video capture, object detection and priority selection without speech output.

8. Experimental Evaluation and Results

The prototype was evaluated using the configured laptop, webcam and software version under four lighting and scene conditions. Table 4 reports the recorded detection accuracy and average response time for 50 tested frames in each condition.

8.1 Environmental Conditions

Fifty camera frames were tested for each condition, where a detection was counted as correct when the reported object label matched the observed object.

The tests covered indoor bright light, indoor low light, outdoor daylight and crowded or cluttered scenes.

For every test frame, the system recorded detection correctness and response time. The priority alert was also checked against the object most relevant to the camera view.

If users participate, obtain institutional approval and informed consent before collecting personal data or publishing feedback.

Environmental condition	Frames tested	Correct detections	Detection Accuracy	Average response time (ms)
Indoor - bright light	50	50	100%	350ms
Indoor - low light	50	50	100%	353ms
Outdoor - daylight	50	50	100%	350ms
Crowded / cluttered scene	50	47	94%	355ms

Table 4. Prototype test results under different conditions.

8.2 Evaluation Metrics

The performance of an object detection system can be evaluated using several standard metrics. These metrics help to measure the correctness of object detection, the ability to detect actual objects, and the accuracy of the predicted bounding boxes. The major evaluation metrics used in object detection are Precision, Recall, F1-Score, Intersection over Union (IoU), and Mean Average Precision (mAP).

1. Precision

Precision measures the proportion of correctly detected objects among all the objects predicted by the system. It indicates how many of the detected objects are actually correct.

Formula:

$$Precision = \frac{TP}{TP + FP}$$

Where:

- **TP (True Positive):** The object is correctly detected.
- **FP (False Positive):** The system detects an object incorrectly.

A higher Precision value indicates that the system produces fewer incorrect detections.

2. Recall

Recall measures the ability of the system to detect all the actual objects present in an image or video frame. It indicates how many of the existing objects are successfully detected.

Formula:

$$Recall = \frac{TP}{TP + FN}$$

Where:

- **TP (True Positive):** The object is correctly detected.
- **FN (False Negative):** The actual object is present but the system fails to detect it.

A higher Recall value indicates that the system is able to detect more of the objects present in the scene.

3. F1-Score

F1-Score is the harmonic mean of Precision and Recall. It provides a balanced measure when both correct detection and complete detection are important.

Formula:

$$F1\text{-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

A higher F1-Score indicates a better balance between Precision and Recall.

4. Intersection over Union (IoU)

Intersection over Union is used to measure the accuracy of the predicted bounding box by comparing it with the ground-truth bounding box. It calculates the amount of overlap between the predicted and actual bounding boxes.

Formula:

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

The IoU value ranges from 0 to 1. A value closer to 1 indicates a greater overlap between the predicted bounding box and the ground-truth bounding box.

5. Mean Average Precision (mAP)

Mean Average Precision is a commonly used metric for evaluating object detection systems. It calculates the average precision across all detected object classes.

Formula:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

Where:

- **N:** Number of object classes.
- **AP_i:** Average Precision of the i^{th} object class.

mAP provides an overall measure of the detection performance of the model across different object classes. Common values include [mAP@0.5](#) and [mAP@0.5:0.95](#), which use different Intersection over Union thresholds.

6. Accuracy

Accuracy represents the proportion of correct predictions among all predictions made by the system.

Formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- **TP:** True Positive.
- **TN:** True Negative.
- **FP:** False Positive.
- **FN:** False Negative.

Although Accuracy is a common classification metric, Precision, Recall, F1-Score, IoU, and mAP are generally more suitable for evaluating object detection systems because object detection involves both object classification and localization.

8.3 Experimental Output

The developed system was tested using live webcam input in different indoor environments. The experimental outputs demonstrate the ability of the proposed system to detect objects, identify their relative position, determine their priority, and provide an apparent proximity category. The detected objects are displayed using bounding boxes and confidence scores, while the dashboard presents the priority obstacle and corresponding environmental information.

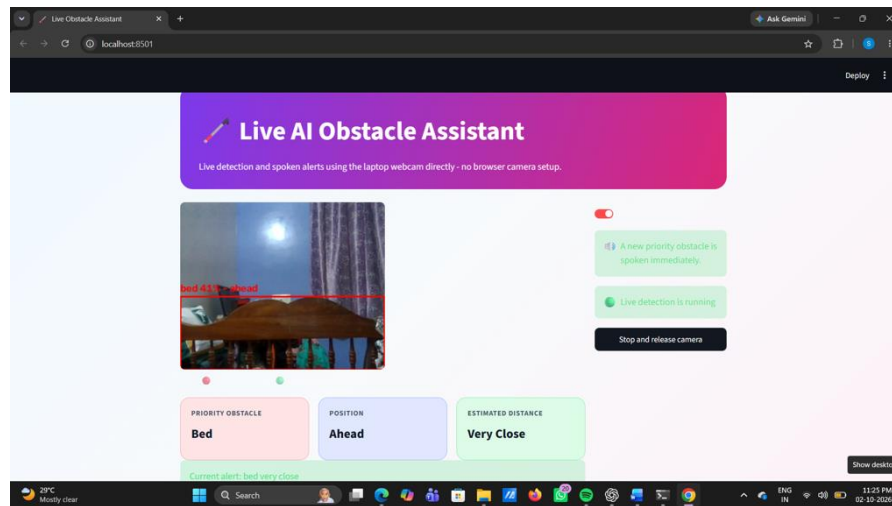


Figure 1. Live detection of a bed with the priority obstacle classified as ahead

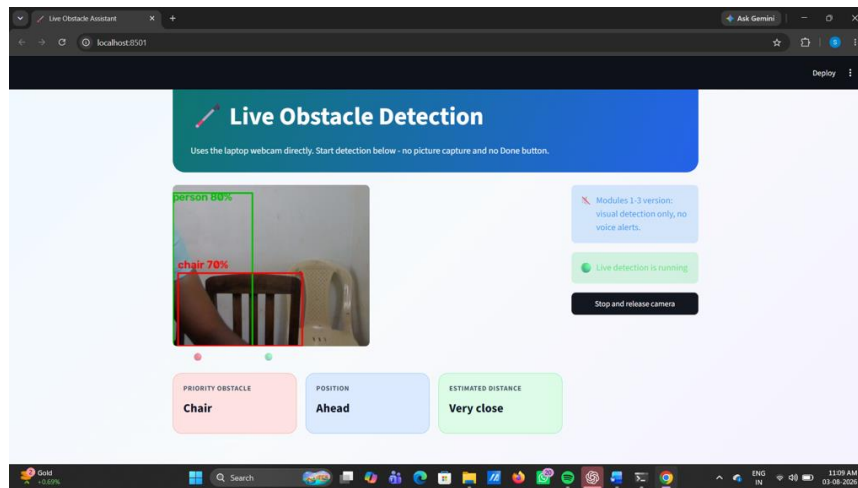


Figure 2. Detection of a person and chair, with the chair selected as the priority obstacle.

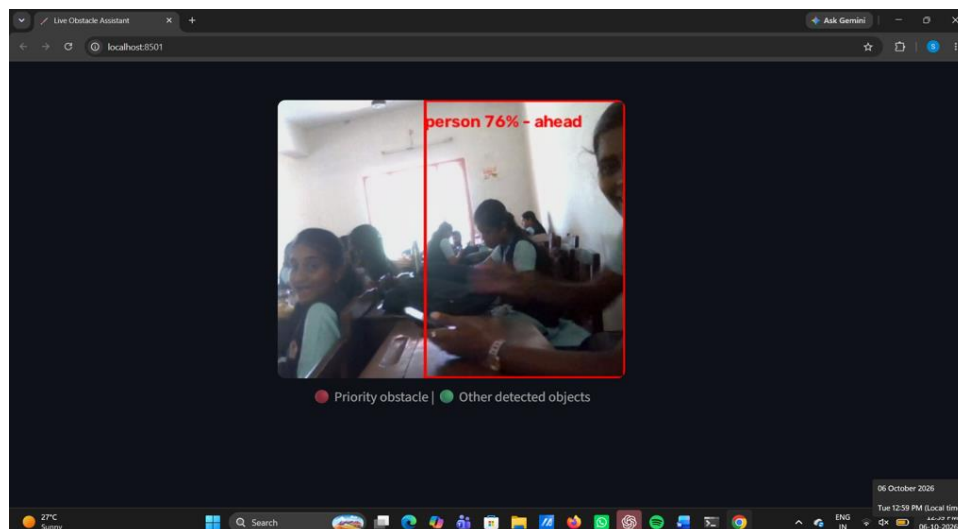


Figure 3. Object detection in a classroom environment .

8.4 Discussion of Experimental Results

The experimental outputs show that the system can detect multiple objects from live webcam frames and provide additional information beyond the object class. In the first example, the system identifies a bed as the priority obstacle and classifies it as ahead and very close. In the second example, both

a person and chair are detected, with the chair selected as the priority obstacle. The third example demonstrates detection in a crowded classroom environment, where a person is identified ahead of the camera. These results demonstrate the practical operation of the object detection, position analysis, priority selection, and apparent proximity components of the proposed system.

9. Discussion

The proposed system combines real-time object detection with a safety-oriented priority method that highlights important obstacles. Open-vocabulary detection allows the system to identify navigation-related objects without training a separate model for every label.

However, the monocular webcam provides only apparent proximity estimates, not accurate physical distances. Detection may also be affected by lighting, occlusion, motion blur, object appearance, and hardware performance. Therefore, the system is an additional awareness aid and does not replace a white cane, guide dog, or safe navigation practices.

10. Conclusion

This paper presented an AI-based real-time obstacle detection assistant for visually impaired people. The proposed system processes webcam video using OpenCV and YOLO-World, selects a priority obstacle according to its position and relative image size, and provides spoken alerts through Windows Text-to-Speech. A Streamlit dashboard shows the live camera output and detection details. The design is low cost because it can run on a standard computer with a webcam and speaker. The prototype test results provide initial evidence of detection accuracy and response time under different environmental conditions; a larger labelled evaluation is needed before broad performance claims are made.

11. Future Work

The proposed system can be further enhanced to provide more accurate, reliable, and user-friendly assistance for visually impaired people. Depth cameras, stereo vision, ultrasonic sensors, or LiDAR can be integrated to measure the actual distance of objects instead of relying only on apparent proximity from bounding-box size. Object tracking can also be implemented to distinguish newly detected obstacles from the same objects appearing in consecutive frames and thereby reduce repeated alerts. The detection model can be trained or fine-tuned using local indoor and outdoor obstacle datasets containing objects such as doors, steps, curbs, poles, vehicles, and other commonly encountered objects. Optical Character Recognition (OCR) can be added to read text from signboards, documents, labels, and other printed materials, while currency recognition can assist users in identifying different denominations. The system can also be extended with GPS-based navigation and route guidance to support independent outdoor movement. Multilingual speech support can be introduced to provide alerts in different regional languages. In addition, voice commands can allow users to interact with the system without using a keyboard or screen. Object tracking, obstacle direction estimation, and improved priority selection can further enhance real-time alerts. The system can also be optimized for mobile devices, wearable devices, or smart glasses, making it more convenient for daily use.

References

- [1] P. Kathiria, S. H. Mankad, J. Patel, M. Kapadia and N. Lakdawala, 'Assistive Systems for Visually Impaired People: A Survey on Current Requirements and Advancements,' *Neurocomputing*, vol. 606, 2024, Article 128284, doi: 10.1016/j.neucom.2024.128284.
- [2] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, 'You Only Look Once: Unified, Real-Time Object Detection,' *Proceedings of CVPR*, pp. 779-788, 2016.
- [3] T. Cheng et al., 'YOLO-World: Real-Time Open-Vocabulary Object Detection,' *Proceedings of CVPR*, 2024.
- [4] H. Najm, K. Elferjani and A. Alariyibi, 'Assisting Blind People Using Object Detection with Vocal Feedback,' *arXiv:2401.01362*, 2024.
- [5] K. Bamane, S. Khaparde, H. Totala, Y. Singh and D. Ghule, 'Design and Development of Android Application for Visually Impaired People Using Object Detection and Direction Prediction,' *International Journal of Novel Research and Development*, vol. 9, no. 4, 2024.
- [6] M. Joshna, J. T. Kumar, G. Yuvaraju, K. V. V. Reddy and V. S. P. Reddy, 'Object Detection Tracking and Alert System for Visually Impaired Persons,' *Proceedings of ICRDICT*, pp. 45-49, 2025.
- [7] A. Nabila, F. A. Soelistianto and I. Mahfudi, 'Designing an Object Detection System as an Assistive Device for the Visually Impaired Based on YOLOv10 with Dual Camera,' *West Science Information System and Technology*, vol. 3, no. 3, pp. 166-175, 2025.
- [8] Ultralytics, 'YOLO-World Model Documentation,' accessed August 2026. Available: <https://docs.ultralytics.com/models/yolo-world/>
- [9] M. M. Valipoor and A. de Antonio, "Recent trends in computer vision-driven scene understanding for VI/blind users: A systematic mapping," *Universal Access in the Information Society*, 2022, doi: 10.1007/s10209-022-00868-w.
- [10] K. Patel and B. Parmar, "Assistive device using computer vision and image processing for visually impaired; review and current status," *Disability and Rehabilitation: Assistive Technology*, vol. 17, no. 3, pp. 290–297, 2022, doi: 10.1080/17483107.2020.1786731.

-
- [11] A. Voulodimos, N. Doulamis, A. Doulamis and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, vol. 2018, Article 7068349, 2018, doi: 10.1155/2018/7068349.
- [12] Z.-Q. Zhao, P. Zheng, S.-T. Xu and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019, doi: 10.1109/TNNLS.2018.2876865.
- [13] G. Jocher, A. Chaurasia and J. Qiu, "Ultralytics YOLOv8," version 8.0.0, Ultralytics, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [14] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.
- [15] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.
- [16] W. Liu et al., "SSD: Single shot multibox detector," *European Conference on Computer Vision (ECCV)*, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.
- [17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, 2005, doi: 10.1109/CVPR.2005.177.
- [18] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.