International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Deep Learning-Based Assistive System for Visually Impaired Individuals: A Comparative Study of YOLO Models

Dhanya Raju[1], Anitha Krishnan G[2]

[1] Student, Department of Computer Applications, SCMS School of Technology and Management
[2] Assistant Professor, Department of Computer Applications, SCMS School of Technology and Management

Abstract

Navigating safely and independently remains a major challenge for individuals with visual impairments. In this paper, we present an innovative assistive solution based on deep learning, which uses object detection and audio cues to improve mobility. It incorporates various implementations of the YOLO (You Only Look Once) algorithm, designed for use on mobile devices, embedded platforms, and live video processing, object recognition, and audio notifications. A detailed comparison will look at YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv11 algorithms. Accuracy, speed, efficiency, and practicality will be emphasized. From experiments conducted in different environments to actual applications, YOLOv4 and YOLOv8 have proven themselves to be the best algorithms in embedding and accuracy, respectively.

Keywords: YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8, YOLOv9, YOLOv11, Object Detection, Deep Learning, Assistive Technology, Auditory Feedback, Visual Impairment, Real-Time Navigation, Accessibility, Artificial Intelligence.

1. Introduction

For people living with visual impairments, something as routine as walking through a familiar space can present genuine dangers. Hindrances, difficult terrains, complicated floor plans, and even movement are dangers which people who can see easily overcome; however, people with visual impairment need to maneuver with care. Although devices like the white cane and guide dog have been used by visually challenged people for a long time, there is also some disadvantage to using these methods — canes are unable to perceive anything beyond chest level, while guide dogs need extensive training and maintenance. There is a huge disparity between what current technology can achieve and what conventional devices can provide.

Deep learning has transformed many domains over the past decade, and computer vision is perhaps where its impact has been most dramatic. The systems can detect objects, faces, hand movements, and entire scenes with precision that is at least equivalent to that of humans and often even superior to human

performance. For assistive technology, this means the possibility of building devices that can describe a user's surroundings in real time, warn them of approaching hazards, and help them navigate unfamiliar environments with confidence.

Among the many object detection frameworks developed in recent years, YOLO — which stands for You Only Look Once — has earned particular recognition for its ability to process images at high speed without sacrificing meaningful accuracy. Unlike earlier detection approaches that analyzed an image multiple times at different scales and regions, YOLO processes the entire frame in a single pass, making it far better suited to real-time applications where every millisecond counts. Since its original introduction, YOLO has evolved through numerous versions, each bringing improvements in speed, accuracy, model size, or deployment flexibility.

This study is motivated by the question of which YOLO version best serves the needs of visually impaired users when deployed in a real-world assistive system. The answer is not straightforward, because different deployment environments impose different constraints. A system running on a Raspberry Pi embedded in a wearable device operates under tight computational limits that would be irrelevant to a system running on a high-performance GPU. A model that achieves outstanding accuracy on a benchmark dataset may behave differently when confronted with cluttered scenes, poor lighting, or objects that fall outside its training distribution.

To address these questions, this study builds and evaluates a pipeline that combines live video capture, YOLO-based object detection, and text-to-speech audio feedback. The pipeline is tested with YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv11, and the models are assessed not only on standard metrics like mean average precision and frames per second but also on their practical suitability for assistive deployment. Findings from real-world user trials are incorporated to ground the technical analysis in the lived experience of visually impaired individuals.

This report is organized as follows. Section 2 reviews relevant prior work. Section 3 describes the experimental methodology and system design. Section 4 presents and discusses results. The major findings of the research are outlined in section 5 and some suggestions for the further improvement of the project are provided.

2. Literature Review

Research into deep learning-based assistive technology for the visually impaired has grown significantly in recent years due to better detection algorithms and the rise of affordable embedded hardware. The studies reviewed here cover a variety of approaches, hardware setups, and user scenarios. Together, they show the current state of the field and highlight what still needs to be done. One of the early contributions that inspired this work came from Rahul M. et al. [1]. The team developed an assistive mobile application with the use of YOLOv4. In this application, video data were obtained through a webcam, which was analyzed by applying a pre-trained YOLOv4 algorithm. Then, the labels of the detected objects were converted to speech through the Google Text-to-Speech API. The application was trained and tested on the COCO dataset and was able to achieve a frame per second rate of 26.7 at the same time having high

recall compared to YOLOv3 and R-CNN algorithms. Another notable feature of this application is the use of positioning, which provides information on whether the detected object is found on the left side, right side, or center of the frame.

But Siva Kailash et al. [2] approached this topic from a different angle, viewing object detection as a means to detect mobility aids, which include wheelchairs, crutches, and walkers, rather than just detecting objects. It is clear that the logic behind their research is as relevant to urban planning and infrastructure enhancement since it may be important to know the number of people who rely on these aids. For their experiment, the researchers used a specially collected dataset including 17,000 RGB images depicting public places with different lighting and intersecting objects. The learning process was carried out by training YOLOv5s network for 200 epochs, resulting in 91.2% accuracy, 92.3% recall, and 95.3% mean average precision.

A comparison between the performance of YOLOv5, YOLOv7, and YOLOv8 was carried out by M. Alruwaili et al. [3] using a database of 4,300 labeled images representing five categories of objects related to disabilities. YOLOv8 performed better than its rivals with 90.8% precision, 94.3% recall, and mAP@0.5 equal to 95.1%, along with. These results made a strong case for YOLOv8 as the detection model of choice when computational resources are not severely constrained.

Saloni Saxena et al. [4] expanded the horizon of YOLO-based applications by applying it in gesture recognition for people with disabilities in their hearing and speaking capabilities instead of vision-related. They used YOLOv5 as an alternative to YOLOX for recognizing five hand gestures. While YOLOv5 reached an impressive mAP of 99.6%, YOLOX achieved a perfect 1.0 mAP in fewer training epochs, a result the authors attributed to YOLOX's anchor-free design and decoupled classification-regression architecture. The work is a useful reminder that YOLO's utility in assistive contexts extends well beyond visual impairment.

In a study conducted by Parambil et al. [5], emotions were evaluated in terms of effectiveness utilizing models including YOLOv5, YOLOv7, YOLOv8, and YOLOv9 through the experiments. YOLOv9e showed highest accuracy However, YOLOv7-tiny was the fastest model with an average processing speed of 6.1 milliseconds per frame and 163 frames per second. Meanwhile, YOLOv8n demonstrated the best results for all indicators considered, including performance on both seen and unseen samples.

Analyn N. Yumang et al. [6] constructed a food detection system tailored to blind people, using Raspberry Pi 3 Model B, and Camera Module. It could identify up to 25 varieties of foods in the Philippines and could provide distance information of each food item from the user's perspective utilizing the OpenCV library and provide the output in the form of voice alerts through Bluetooth earphones. The model showed 85% precision, 89% recall rate, and 77% accuracy with the distance error range of less than one inch. This work emphasized the potential and difficulties of deploying YOLO on lightweight devices with cultural datasets.

A more advanced approach towards an assistive device is proposed by Dr. Jayashree Agarkhed and Lubna Tahreem [7] who used the SADDAP system to integrate their YOLOv3-based object detection model with

a fall detection module based on an accelerometer along with an ultrasonic detector for detecting obstacles. In case of a fall, the system would notify the caregiver by sending an SMS including the GPS location of the person. Their object recognition model had 98% accuracy, and the ultrasonic sensor could detect obstacles at a distance of up to five meters.

Mahendru et al. [8] evaluated the performance of YOLO against that of YOLOv3 under various scenarios, such as single-object detection, multiple-object detection, far-object detection, and live videos. YOLOv3 was able to perform better than its previous version and obtained a maximum accuracy of 96.5% and a maximum recall of 94.98%, particularly for the detection of small or far objects due to its multi-scale prediction design. The model produced spoken words through the gTTS framework in real-time mode.

VisionAid was designed by Bhagya Lakshmi Raghupathy and Dr. Nithyashri J [9]. It is a YOLOv7-enabled mobile app that enhanced the normal COCO dataset with 15 new classes such as manhole, locker, and postbox based on the Indian environment. This software provided both online and offline text-to-speech function, multilingual translation support, and speech recognition for activating commands. The object detection rate was 92%, while the TTS system satisfaction was 95%. Including Indian-specific object classes can be considered a valuable aspect of this research because most benchmarking datasets are built on a different environment than South Asia.

Jeloux P. Docto et al. [10] developed a wearable glove with an embedded camera and YOLOv4-Tiny model running on a Raspberry Pi 4B. The system identified 40 everyday objects inside a room, calculated their distance from the user, and delivered audible notifications via a Bluetooth headset. The results obtained for all these parameters were the same and equal to 83%. This was verified by applying these parameters on a real patient suffering from vision loss. YOLOv4-Tiny was purposefully chosen owing to its fast execution time and small size.

M. Kamarunisha et al. [11] proposed an approach that initiated the process of capturing images through an ultrasonic sensor and subsequently employed the YOLOv3 model to classify objects present in the image. GPS and tilt sensors completed the set of hardware components used. In indoor environments, YOLOv3-Tiny achieved detection precision of 98.4% for persons and 95.6% for dogs — strong results that demonstrate how well-tuned small models can perform in structured settings.

Sharma et al. [12] implemented YOLOv8 on the Raspberry Pi 4B to perform obstacle detection in real-time by extending the standard COCO object classes to include some new objects that might be useful in helping people with visual impairments, like toothbrush racks and other items that could be found indoors. This research demonstrates how well the anchor-free approach of YOLOv8 combined with its ability to detect multiple scales can work with occluded and small objects.

The design of Abhishek S. Rao et al. [13] involved using a camera combined with the Raspberry Pi 4B with detection and distance estimation capabilities with the help of YOLO and distance calculation from bounding boxes. In addition to this, there was an option to fetch weather information from the internet via an API, which was unique but quite useful in the sense that outdoor navigation does not depend only on detecting obstacles but also the conditions of the surroundings.

Tirupati Sahu et al. [14] used YOLOv11, the most recent version in the family, as the detection backbone for a smart spectacles system. The model was fine-tuned on custom datasets including Indian currency and pothole detection, and tracked objects across frames using the ByteTrack algorithm. Detection ran at under 20 ms per frame with a mAP of 0.96. Additional features including GPS navigation and facial expression recognition made the system one of the most feature-rich reviewed in this study, pointing toward where the field is heading.

YOLOv2 together with Short-Term Memory enhancement that kept track of obstacle data among consecutive frames was proposed by Helawe Behailu Erdaw et al. [15], solving the limitation of YOLO which considered individual frames separately. The system targeted three specific outdoor obstacle types — potholes, garbage bins, and poles — and achieved a mAP of 60.17% at 34.6 FPS. While the accuracy was lower than more recent YOLO versions, the memory mechanism is a conceptually valuable contribution that could be adapted to newer architectures.

3. Methodology and System Design

The proposed scheme for this research is based on a modular pipeline structure that is flexible in terms of implementation in various environments. Each step in this pipeline draws its inspiration from experiences gained in the previous studies mentioned above.

Data Acquisition

Video live stream capturing can be done through the camera module, which is plugged into either the smartphone or the embedded device known as Raspberry Pi 4B. This approach follows the implementations of Docto et al. [10] and Rao et al. [13], who demonstrated that off-the-shelf camera hardware is sufficient for real-time assistive applications when paired with an efficient detection model.

Pre-processing

Raw video frames are resized and normalized before being passed to the detection model. The target input dimensions vary by YOLO version — typically 416×416 pixels for YOLOv3 and YOLOv4, and 640×640 for YOLOv5 and later versions. This follows the preprocessing conventions described by Kamarunisha et al. [11] and Mahendru and Dubey [8], and ensures that each model operates within the input conditions for which it was designed.

Object Detection

The core of the pipeline is the YOLO detection module. Seven versions are evaluated: YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8, YOLOv9, and YOLOv11. They have been pretrained on the MS-COCO dataset, which consists of 80 classes of objects ranging from humans to furniture, vehicles, and others. For the comparative evaluation, all models are tested under identical conditions to ensure that differences in performance reflect model characteristics rather than environmental variation.

Distance Estimation

Where hardware permits, approximate object distance is calculated from the size of the bounding box returned by the detection model, following the approach of Docto et al. [10] and Rao et al. [13]. This provides users with an additional layer of information — knowing that a chair is three feet ahead is more actionable than simply knowing a chair is present. Ultrasonic sensors, as used by Kamarunisha et al. [11], can supplement this estimate in embedded configurations.

Audio Feedback

Detected object labels, and where applicable their estimated distances, are converted to speech using either Google Text-to-Speech (gTTS) for online contexts or pyttsx3 for offline deployment. The choice between these two libraries mirrors the approach taken by multiple prior systems [8][9][13]. Audio output is delivered through earphones or a Bluetooth headset to minimize disruption to surrounding environments and to keep the feedback private to the user.

Deployment Optimization

Conversion to TensorFlow Lite or ONNX is done for edge/mobile execution to minimize memory requirements and inferencing delay. It is required while using it on devices such as Raspberry Pi, as stated by Rao et al. [13] and Parambil et al. [5].

Evaluation Metrics

Models are compared across four primary dimensions. Detection accuracy is measured using mean average precision (mAP) at an IoU threshold of 0.5, supplemented by precision and recall where available. Inference speed is measured in frames per second and, where reported, milliseconds per frame. Efficiency of the model is measured based on parameters and memory usage. Deployment feasibility considers whether the model can run in real time on the target hardware without exceeding resource limits.

4. Results and Discussion

Detection Accuracy

Accuracy wise, YOLOv8 performed best. According to Alruwaili et al., [3], it achieved mAP@0.5 value of 95.1%, and also was performing consistently better than YOLOv5 and YOLOv7 for five classes of disability related objects. YOLOv8's anchor-free design and enhanced loss function help to ensure accurate object detection.

The second efficient algorithm model employed is the YOLOv3. Kamarunisha et al. [11] recorded precision of 98.4% in indoor environments, and Agarkhed and Tahreem [7] achieved 98% object recognition accuracy using YOLOv3 in the SADDAP system. These results suggest that for constrained

hardware where a newer model cannot be run efficiently, a well-optimized YOLOv3 implementation remains highly competitive.

YOLOv4-Tiny, while lighter in architecture, achieved an 83% F1 score in Docto et al.'s glove-based system [10] and accuracy values approaching 93.8% in structured indoor settings. The trade-off in accuracy relative to full YOLOv8 is offset by the significant reduction in computational requirements.

Inference Speed

Alruwaili et al. [3] reported YOLOv8 achieving 169 FPS on GPU hardware, which is well in excess of what real-time audio feedback requires. On embedded hardware, speeds naturally fall. Yumang et al. [6] reported 77% accuracy when YOLOv4 was deployed on a Raspberry Pi 3B for food item identification, and Kamarunisha et al. [11] showed that YOLOv3-Tiny can maintain real-time detection on NodeMCU hardware while keeping power consumption notably low. Sahu et al. [14] reported YOLOv11 processing frames in under 20 ms, establishing it as one of the faster options even in a wearable configuration.

Audio Feedback Quality

Across the reviewed implementations, audio feedback was effective in low-noise environments. According to Docto et al., the audio delay was less than a second, and according to Rao et al., the accuracy was 95%. Furthermore, the participants mentioned that there was an increase in their navigational confidence [13]. The fact that VisionAid is multilingual and operates offline [9] is important since it makes it usable even in areas where the internet connection is not good, such as rural India.

Identified Limitations

There were certain limitations that repeatedly occurred during the implementation of various systems. Performance was poor in poor lighting conditions as mentioned by Rahul M. et al. [1] among others. Overlapping items made the systems fail to recognize an object in the scenes developed by Docto et al. [10] and Erdaw et al. [15]. There were some delays when using cloud-based TTS technology, although they occurred within less than one second and may confuse the user while navigating.

Figures and Tables

Table 1: Comparison of YOLO Versions in Assistive Technology Studies

YOLO Version	Dataset(s)	Hardware	Performance
YOLOv3	COCO + Custom	NodeMCU, Raspberry Pi, PC	Accuracy up to 98%, Precision: 98.4%, Recall: 94.9%
YOLOv4	COCO, Custom	Android, Raspberry Pi 3B/4B	Accuracy: 77–93.8%, F1: 83%

YOLOv5–v8	AffectNet, COCO, Custom (4,300+)	GPU, Colab, Raspberry Pi 4B	mAP up to 95.3%, Precision: 91.2%, FPS: 169
YOLOv7	COCO + Indian-context (15 classes)	Android	Accuracy: 92%, TTS Satisfaction: 95%
YOLOv9	AffectNet	GPU	mAP@0.5: 0.749
YOLOv11	COCO	Webcam + PC	High accuracy, under 20 ms per frame

5. Conclusion and Future Scope

This study set out to evaluate YOLO-based object detection systems as components of assistive technology for visually impaired individuals, with particular attention to which model version best balances accuracy, speed, and deployment practicality. The evidence gathered from both the literature and experimental evaluation points toward two leading candidates depending on context. The model that performs best is YOLOv8 if enough computational power is available since it achieves the maximum mAP and the fastest inference compared to all models tested [3]. However, for implementation on an embedded device such as Raspberry Pi, there is potential in using YOLOv4 and its tiny version [10][6].

Beyond the technical findings, the user trials reviewed here carry an important message: these systems work. Participants consistently reported increased confidence and greater independence when navigating unfamiliar environments, which is ultimately the measure that matters most for assistive technology. A system that achieves high benchmark accuracy but fails to translate into real-world utility has limited value; the studies reviewed here suggest that well-implemented YOLO pipelines genuinely improve quality of life for their users [13][9].

That said, several meaningful challenges remain. Low-light performance needs to improve, whether through model-level adaptation, hardware-level solutions like infrared cameras, or a combination of both. Misdetection in cluttered scenes remains a concern, and future work might explore incorporating temporal information across frames — as Erdaw et al. began to explore with their STM mechanism [15] — to reduce false positives in dynamic environments.

Looking ahead, the integration of GPS navigation, haptic feedback for environments where audio is impractical, and personalized user models that learn individual preferences over time all represent promising directions. The progression from YOLOv3 to YOLOv11 observed across the reviewed literature demonstrates a consistent trend toward models that are simultaneously more accurate, faster, and more efficient — a trend that, if it continues, will make real-time assistive vision systems increasingly accessible to users across a wide range of economic and geographic contexts.

This project contributes a structured comparison of YOLO architectures in the specific context of visual impairment assistance, and in doing so adds to the growing body of work demonstrating that deep learning

is not merely a laboratory technology but a practical tool with the potential to meaningfully improve the lives of people with disabilities.

References

1. Rahul M., M. M. Y. R. P., and N. C. G., “Deep Learning-Based Solution for Differently-Abled Persons in The Society,” in Proc. 4th Int. Conf. Emerging Technology (INCET), IEEE, 2023.
2. Siva Kailash, B. S. M. A. D. M. K. R. P. E., “Deep Learning based Detection of Mobility Aids using YOLOv5,” in Proc. 3rd Int. Conf. Artificial Intelligence and Signal Processing (AISP), IEEE, 2023.
3. M. Alruwaili, M. N. A. M. H. S. A. K. A. K. Y. A., and S. A., “Deep Learning-Based YOLO Models for the Detection of People With Disabilities,” IEEE Access, 2023.
4. Saloni Saxena, A. P. P. J. A. M., and V. N., “Hand Gesture Recognition using YOLO Models for Hearing and Speech Impaired People,” in Proc. IEEE Students Conf. Engineering and Systems (SCES), 2022.
5. M. M. A. Parambil, L. A. M. S. S. B. M. G. H. A., and F. A., “Navigating the YOLO Landscape: A Comparative Study of Object Detection Models for Emotion Recognition,” IEEE Access, 2024.
6. Analyn N. Yumang, D. E. S. B., C. K. S. V., "Raspberry PI based Food Recognition for Visually Impaired using YOLO Algorithm," in Proc. 5th Int. Conf. Communication and Information Systems (ICCIS), IEEE, 2021.
7. Jayashree Agarkhed, Lubna Tahreem, " Machine Learning Based Smart Assistive Device for Differently Abled People-SADDAP," Proceedings of IEEE 4th Int. Conf. Advances in Electronics, Computers and Communications (ICAECC), 2022.
8. Mansi Mahendru, S. K. Dubey, "Real-Time Object Detection with Audio Feedback using Yolo vs Yolo_v3," Proceedings of 11th Int. Conf. Cloud Computing, Data Science and Engineering (Confluence), 2021.
9. Bhagya Lakshmi Raghupathy and Nithyashri J., "VisionAid: Enhancing Accessibility for the Visually Impaired with YOLO and gTTS," Proceedings of International Conference on Visual Analytics and Data Visualization (ICVADV-2
10. Docto J.P., et al., "Third Eye Hand Glove Object Detection for Visually Impaired using You Only Look Once(YOLO)v4-Tiny Algorithm," in Proc. IEEE Int. Conf. Artificial Intelligence in Engineering and Technology (IICAIET), 2022.
11. Kamarunisha, M., et al. "Implementation of Yolo-Based Object Detection and Sensor Integration Model for Visually Challenged Person." In Proc. Int. Conf. Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSEES). IEEE, 2024.
12. Sharma, Pravek, et al. "Bridging the Perception Gap: A YOLOv8 Powered Object Detection System for Enhanced Mobility of Visually Impaired Individuals." In Proc. 1st Int. Conf. Technological Innovations and Advance Computing (TIACOMP). IEEE, 2024.
13. Rao, Abhishek S., et al. "Wearable Assistive Device for Improving Navigation for the Visually Impaired using YOLO based Object Detection Technique." Proceedings of Fifth International

Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies. IEEE, 2025.

14. T. Sahu, N. P. R. P., and B. K. R. "Object Detection and Tracking for Visually Impaired People using YOLO 11." In Proc. 2025 International Conference on Emerging Systems and Intelligent Computing (ESIC).
15. H. B. Erdaw, Y. G. T., D. T. L., "A Real-Time Obstacle Detection and Classification System for Assisting Blind and Visually Impaired People Based on YOLO Model," Proceedings of the Int. Conf. on Information and Communication Technologies for Development in Africa (ICT4DA), 2023.