



## Review on YOLO-based Mobility Assistance Systems for the Visually Impaired

B. Abhitha Bhat<sup>1</sup>, Gaurav S B<sup>2</sup>, Shashank Ravindra<sup>3</sup>, Ananya Raghuveer<sup>4</sup>, Nandita Bangera<sup>5</sup>

CSE, RV Institute of Technology and Management,  
Visvesvaraya Technological University

### ABSTRACT

This paper presents a machine learning-based framework aimed at enhancing mobility and spatial awareness for the visually impaired by reviewing different papers where authors suggest various object detection models, particularly focusing on different versions of the YOLO algorithm, to provide real-time analysis of a user's surroundings. The reviewed systems are audio feedback integrated, converting visual data into descriptive audio cues to help users identify objects and obstacles, enabling independent navigation. The paper also reviews the use of the Darknet framework with TensorFlow, and tools such as Google Text-to-Speech (GTTS) and Raspberry Pi, emphasising their role in building effective, real-time assistive technologies.

**Keywords:** YOLO (You Only Look Once), Machine Learning, Raspberry pi, COCO Dataset, TensorFlow, SSD (Single Shot Detection), GTTS (Google text to speech), Darknet.

### 1 Introduction

Vision provides essential information about the world around us and plays a crucial role in our interactions, navigation, and overall well-being. Through sight, people can move independently, recognize objects, and detect obstacles, which fosters a sense of security and confidence. But for those who are visually impaired, the reduction or absence of vision creates unique challenges, greatly impacting their mobility and independence. Everyday tasks—like recognizing objects, navigating rooms, or avoiding obstacles—become complex, often requiring help from others or specialised tools. These limitations can deeply affect their quality of life, inclusion, and sense of autonomy.

Visually impaired individuals face broad and multifaceted difficulties. Moving safely and confidently through unfamiliar spaces is a significant challenge, as the inability to see heightens the risk of disorientation and injury. Identifying objects is challenging—whether in public, where they may struggle to locate entrances, navigate steps, or detect vehicles, or at home, where distinguishing between similar objects, like bottles, becomes difficult without visual cues. Without real-time information about their surroundings, visually impaired individuals often rely on others for support, limiting their independence.

Recent advancements in computer vision and deep learning offer promising solutions for increasing autonomy. Object detection, a specialised area of computer vision, focuses on recognizing and locating objects in images or video. By integrating object detection with voice-assisted technology, users can receive real-time audio feedback about nearby objects, helping



them interpret and engage with their environment independently. This powerful combination enhances mobility and situational awareness, creating a safer, more inclusive experience for users.

Advanced algorithms in object detection research have led to techniques like You Only Look Once (YOLO) V4 and Single Shot MultiBox Detector (SSD), which provide the accuracy and speed needed for real-time recognition. These models can detect multiple objects and track their positions within a live video feed, making them highly suitable for assistive applications. Coupled with voice synthesis software like pyttsx3, these systems provide continuous spoken feedback on detected objects, helping visually impaired users stay aware of their surroundings. The Darknet neural network framework, an open-source toolkit developed in C and CUDA, offers a critical foundation for implementing object detection models. Known for its flexibility and efficiency, Darknet powers models like YOLO, setting a new standard in real-time object detection. This framework allows rapid processing of visual data, making it ideal for assistive technologies aimed at improving mobility and independence for visually impaired individuals. Additionally, Darknet's open-source nature encourages collaboration and ongoing innovation, inviting contributions from researchers and developers worldwide who are committed to advancing object detection systems for real-world applications.

## 2 Literature Review

In [1], the author merges the benefits from YOLO with the help of GTTS for complete detection and communication. Based on inputs received from cameras, as well as LiDAR sensors, the system applies COCO annotated format for using YOLO v3 where the localization is processed from Darknet. Hybrid approaches combine both reactive and deliberative methods by incorporating sensor-based immediate detection and map-based path planning. In improving obstacle detection through YOLO, various loss functions exist. These are L2 Loss, Huber Loss, and Smooth L1 Loss.

In [2], this study mentions use of the YOLO V4, the optimised accuracy version with the help of pyttsx3 to enable real-time object detection and speech conversion. It is using the backbone CSPDarkNet53 that makes YOLO V4 efficient for object recognition. This web app, powered by the open-source framework Streamlit, allows for real-time as well as image detection of objects, further increasing the user's engagement in machine learning. In the case of camera-based detection, the system uses the SSD model. YOLO V4 can run up to 45 FPS and thus is faster and efficient compared to previous versions and shines at object identification and detection.

In [3], the author states on combining on-device object detection using TensorFlow Lite and the recognition of the phrase "OK GOOGLE," leading to voice response. It will be an assistant for the visually impaired by being able to scan an environment for objects and distances, therefore alerting to the closest. The technologies used were Android Studio, TTS, and Speech Recognizer, with the coding done in Java, Python, and XML. Pruning and quantization capabilities in TensorFlow Lite make it suitable for any smartphone camera for optimised



performance. Using a 60% confidence level, the app then defaults and waits for 3 seconds after each alert, making it very fast, friendly, and cheap.

In [4], the authors have developed a walking aid that enables the navigation of the environment around a person without bumping into any object. It is a three-layer system in which the first layer provides reliable obstacle detection through TensorFlow pre-trained models. It then converts the information into speech output through gTTS, informing the users of objects around them. The final layer comprises an Arduino Uno along with a gyroscope, accelerometer, and two servo motors to keep the camera stabilised, front-facing. This innovative gadget gives audio feedback, which keeps user hands free for easier navigation in crowded or uneven areas. In [5], the system proposed by the author utilises a Raspberry Pi that runs on 5V DC and has the better image processing capability with size and mobility limitations that the existing systems have. The proposed system uses the ROI, Region of Interest method for text localisation and recognition, with a quad-core Broadcom 2711 processor, Wi-Fi, and USB ports. It captures images at 5-pixel clarity and records HD video at 360 fps by the Raspberry Pi camera. It is one that will determine the neural network's inputs in doing object detection. The algorithm further applies regression techniques in computing data point distances to determine what's needed. They used linear regression models with a slope and position component, thus cost-effective yet friendly for the user especially in case of a visually disabled one.

In [6], the author has made use of the TensorFlow Object Detection API for model development and trained the model on the COCO dataset of 330K annotated images. OpenCV and Python 2.7 were used to run the system on a Raspberry Pi 3. It has the ResNet backbone for feature extraction and the SSD head for bounding box and class predictions. Multiple anchor boxes and flexible aspect ratios have also been included in it for better accuracy. Unlike such approaches, depth estimation techniques tend to consider spatial structures. Depth-wise separable convolutions in MobileNet produce bounding boxes.

In [7], in the next paper we have reviewed, we get to see that the project evaluates the technologies used to aid the visually impaired by focusing on OCR (Optical character recognition). The conversion of images to sound is done for identification of objects using techniques like SSD and APIs from TensorFlow to ensure that it goes through processing quickly and fits the conditions. Indoor face recognition is done using the Haar Cascade Algorithm, while the image classification is ensured through TensorFlow and pre-trained models. TSM-KF algorithm is applied to the vision-based mobile navigation so that the effectiveness of the solution can be enhanced even further. The camera is connected through its system architecture to a networked server for processing, with very good accuracy achieved of 98% for object detection, 95% for face detection, and 90% for OCR.

In [8], the study is about a wearable object recognition device by making use of YOLOv4 and the COCO dataset. Based on speed and accuracy, this paper uses YOLOv4-Lite, includes three-stage testing with user satisfaction surveys, obstacle detection with LiDAR and cameras, and in its one-stage detection pipeline of YOLOv4 makes use of CSPDarkNet-53 for enhanced



feature propagation and SPP+PANet for multi-scale features. It is powered by a Raspberry Pi 4, earphones with an integrated sound, and a 7-inch LCD monitor for visual and audio feedback. The portable power bank provides support for autonomous operation. All train and test scenarios are trained in 75% training and 25% testing splits. Detection occurs around one second per image in all cases with 100% success rate.

In [9], the algorithm that the author has used for the recognition of objects is based on YOLOv3 and MS COCO dataset; it achieves a mAP of 57.9% on test-dev, running at a speed of 30 frames per second. It has to be implemented using either OpenVMS or Keras. The feature learning from the input is done through Convolutional Neural Networks and further by Non-Max Suppression to ensure the same object is predicted at a time. Yolo v3 is much faster and more accurate than the previous versions. YOLOv2 had a 30-layer Darknet-19 architecture that was having a problem with small object detection, and it used squared error terms in its loss function, whereas YOLOv3 replaces these terms with cross-entropy terms and achieved an impressive accuracy of 93%, which is four times better than the previous versions.

In [10], we have reviewed that the study uses the EPSON BT-300 smart glasses with the Android OS and TTS API to design a deep learning-based smart glasses system. YOLOv3 is utilised, which, upon capturing an image, saves it to an SD card and transfers it to the server for processing. The COCO dataset is used to train YOLOv3, which recognizes 80 object classes, and the Darknet model is converted into a Keras-compatible format. It boasts 96.3% accuracy. The model includes the Intel Core i7-8700K CPU, NVIDIA GTX 1080 Ti GPU, 16GB RAM, and Ubuntu 16.04 OS for processing images and voice output.

In [11], the author makes use of the popular YOLO i.e, the author has used YOLOv4 which classifies images in real-time and predicts distance using OpenCV with an EBICS system powered by a virtual assistant, Ally, using the pyttsx3 package. YOLOv4-tiny, since it is edge device-centric, runs under the Darknet framework pre-trained on COCO which uses Leaky ReLU activation. The Pi camera module is mounted on custom 3D-printed glasses for visually impaired users. The distance estimation module, developed with OpenCV and HAAR cascade classifier, predicts 90 object classes. It runs on a Raspberry Pi 4 with a Quad-core Cortex-A72 ARM processor at 1.5 GHz and 4GB of RAM. Thus, YOLOv4-tiny has inference time faster than precision required for the detection. It employed depth information from the camera for setting bounds and calculating distance from objects using contour area measurements.

In [12], the author has mentioned that methods of SSD detection are implemented using the TensorFlow Object Detection API, and changes may include a MASK Regional Convolutional Neural Network. The voice generation module is completed using the pyttsx3 library. The system tests image accuracy through APIs, using the SSD algorithm. Although R-FCN and SSD models are relatively faster, they do not even come close to the level of accuracy of Faster R-CNN, which presents a slight accuracy improvement. A faster R-CNN with Inception ResNet, with 300 proposals, is the most accurate, at 1 FPS. However, for the large objects, SSD scores more than Faster R-CNN and R-FCN due to its lighter extractors and faster speed.



In [13], the author in his research paper discusses a solution for aiding partially sighted people in observing and spotting objects using the Python language, the Google TensorFlow library, Google Colab and the OpenCV computer vision library. It used the direct video streaming of an RGB webcam of  $1024 \times 768$  and kept this problem of object detection out by using the static image. The system has 30 frames per minute reading as it extracts maximal entropy features by pre-processed through images. TensorFlow enables real-time object detection, providing audio alerts for obstacles and voice output every second. The module of object recognition uses CNN to measure distance and combines it with Google's Text-to-Speech for voice output, not requiring a very large database, while maximising the accuracy and efficiency of recognition. In [14], the author has mentioned that the algorithms used include YOLO and YOLO\_V3. Audio feedback is accompanied with real-time object detection in an application using the OpenCV library. YOLO, "You Only Look Once" uses a single neural network to divide the image into grids, predicting both the bounding boxes and their respective confidence scores, shaped as (S, S, BX5+C). The author also used TensorFlow for the object detection model with SSD\_MobileNet, in which the class and position were simultaneously detected. YOLO\_V3-106 layers with the increased precision due to detections done on three scales. According to the performance, the performance seems better for small object detections. Further research works in this regard would require a broader extension of exploration through self-exploring datasets. The output of text is transformed into speech by means of gTTS that gets played by pygame as one of the fastest real-time algorithms.

In [15], the author begins by using the YOLOv3 algorithm to analyze an input image, resizing it to  $416 \times 416$  pixels and running it through a 106-layer CNN that includes features like residual and upsampling layers. The CNN produces a tensor containing boundary box coordinates and class probabilities, filtering out boxes with probabilities below 0.5 and refining the results through non-max suppression. The COCO dataset identifies 80 different classes. The image is then enhanced by converting it to grayscale and applying a Gaussian blur with a  $3 \times 3$  kernel. The dimensions for edge detection are configured so that the top and bottom widths align with the original image width, while the heights are set to 25% of the original height. The side widths are adjusted to 25% of the original width, maintaining the full height of the image. The author mentions all images used in his research were sourced from Unsplash[19].

In [16], the author has proposed a system which makes use of the YOLOv8 model. YOLOv8 surpasses earlier YOLO models regarding the aspect of precision and speed with an average Mean Average Precision of 0.82 while an IOU threshold is kept at 0.50. Important features associated with YOLOv8 are its anchor-free detection mechanism, new blocks for convolution, and mosaic augmentation. The system contains a camera inside modified spectacles that captures live images, processes them using YOLOv8 to detect multiple objects of different sizes, and translates these detections into audio outputs. Optical Character Recognition (OCR) and text-to-speech (TTS) technologies make this system even more interesting by translating text into speech, enabling users to find their way around independently. The model is then



trained using custom datasets with Python by using Darknet for object classification and bounding box prediction.

The [17] paper is a review prepared by the author which is about object detection, focusing on deep learning models, and comparing the backbone networks such as AlexNet, ResNet, and CSPDarknet. It reviews versions of YOLO, SSD, RetinaNet, and two-stage models such as R-CNN and Faster R-CNN. Key metrics such as IOU, mAP, and FPS will be discussed along with datasets like PASCAL VOC and MS-COCO. Future Directions: the paper deals with better speed and accuracy, tiny object detection, 3D object detection, few-shot learning, detection through anchors, transformers besides widening the scope of detection through evolutionary methods, which has been presented through simplified charts and tables.

In [18] the author has proposed the use of the YOLO v5 model to detect objects using a camera, and speech generation tools are used to convert text from detected images into speech using gTTS (Google Text-to-Speech) and pyttsx3. The system, to be proposed and titled "Third Eye" for the visually impaired, would employ YOLOv5 to detect images in a customised dataset, which would be the MS COCO 2017 dataset, to generate audio descriptions through pyttsx3 and gTTS. The author has mentioned that YOLOv5 uses about 90% less space compared to YOLOv4 but provides comparable accuracy while giving better performance and a lighter framework. The model was trained using a Tesla K80 GPU with 12 GB RAM in Google Colab for 50 epochs with a batch size of 8.

### 3 Results and Discussion

After reviewing the above studies from different authors, we have obtained the following table that summarises the mAP score for different YOLO versions we have come across in our survey.

**Table 1:** Comparison of different YOLO versions

| Algorithm     | MAP score (%) | Advantages                                                                                                                                                                                           |
|---------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| YOLO V1       | 63.4%         | Real-time object detection at high speeds.                                                                                                                                                           |
| YOLO V2       | 78.6%         | YOLO V2 addresses YOLO V1's limitations with multi-scale training, surpassing Faster R-CNN and SSD in real-time efficiency.                                                                          |
| YOLO V3       | 57.9%         | YOLO V3's multi-scale prediction enhances small object detection, making it more efficient than SSD.                                                                                                 |
| YOLO V3- spy  | 60.6%         | Improved accuracy with spatial pyramid pooling for better feature extraction.                                                                                                                        |
| YOLO V4       | 64.17%        | Optimal speed and accuracy of object detection.                                                                                                                                                      |
| YOLO V4- Tiny | 40%           | Improve the model's detection ability for objects of various sizes.                                                                                                                                  |
| YOLO V5       | 48.5%         | Fast, easy to use, and capable of achieving state-of-the-art results for object detection tasks.                                                                                                     |
| YOLO V8       | 53.7%         | Offers real-time, high-precision object detection, robust occlusion handling, and a unified framework for detection and classification, adaptable across diverse hardware from CPUs to edge devices. |

### 4 Conclusion

In summary, it represents an important step in developing assistive technology-a framework on machine learning that supports enhancement of spatial awareness within a visually impaired individual. Through the YOLO algorithm, it uses platforms such as Raspberry Pi and Darknet in processing real-time data relating to the environment for processing into audio cues readily



understood. This ensures that the visually impaired people can move from one place to the other independently without depending on others so much, hence, raising their quality of living. However, there are certain limitations that shall be acknowledged. The system is hardware-dependent and relies on camera resolution and processor speed, which leads to the overheating as well as quick battery drain in mobile applications. Environmental constraints are present such as low-light, cluttered, and crowded spaces, with variable lighting hence causing difficulty in detection of the object. Detection of multi-object remains hard, especially when objects are occluded or uncommon because small datasets lead to accuracy reduction and delay in real-time processing. The findings show the capability of machine learning in really making differences in day-to-day activities of those who are blind or otherwise have impaired eyesight. With the integration of YOLO with audio feedback tools like GTTS and pyttsx3, this "smart stick" has afforded the user a relatively slick experience: not only was the object detection good enough but also provided this to the user in terms of audio, thereby giving better means of interacting with a space. Efficiency and versatility of YOLO variations, such as YOLO V4 and YOLO V4-Tiny, make algorithms a prime candidate for usage within portable assistive devices. This can be viewed as demonstrating the feasibility for the application of this framework towards practical use and even possibly greater applications with the assistive technology. The validation by case studies and a literature review of this new technology as opposed to other previous technologies is what verifies that the framework has better functionality and flexibility. Its additional components, like the COCO dataset, and also the SSD models will significantly enhance the object-people and vehicle recognition with distinction from environmental hazards, further increasing its feasibility toward positively impacting the visually impaired differently at various income levels. This paper shows the existing studies in the field of object detection for the visually impaired using YOLO algorithm while also pointing to exciting areas for future research and development. Future updates could include multi-language support, enhanced text recognition or OCR, and improved obstacle detection such as detection of multiple objects in complex environments. This work, therefore, creates a foundation for practical and inclusive tools that can easily eradicate the day-to-day issues of visually impaired people. As technology advances, so does the possibility of having even more independence and safety for the visually impaired users. In advancing object detection and real-time audio feedback, this study provides groundwork for a more inclusive world where everybody can move with confidence and independence, not based on visual ability.

## 5 References

1. Shriram; Sanjay; Deepika, Year: 2023, "Obstacle Avoidance for blind people using Yolo algorithm, Darknet and GTTS", In ITM Web of Conferences, Vol: 56, pp. 05010.



2. Bindamrutha; Likhitha; Reddy; Reddy; Priya, Year: 2023, "A Real-time Object Detection System for the Visually Impaired", In 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), pp. 1-5.
3. Ephzibah, Year: 2021, "Assisting blind people using machine learning algorithms", Turkish Journal of Computer and Mathematics Education (TURCOMAT) Vol. 12, no: 8, pp. 3162-3170.
4. Hingorani; Mukherjee; Sharma; Mani; Stonier, Year: 2022, "Design and development of a smart blind walking stick using machine learning", Journal of Medical Engineering & Technology, Vol: 46, no: 4, pp. 335-340.
5. Durgadevi; Thirupurasundari; Komathi; Balaji, Year: 2020, "Smart machine learning system for blind assistance", In 2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS); pp. 1-4.
6. Devi; Sahaja; Santrupth; Harsha; Reddy, Year: 2022, "Blind Assistance System using Image Processing". Global Journal of Computer Science and Technology.
7. Karamchandani; Kumari; Dutta; Jalall; Saurav; Kumar, Year: 2023, "Development of Machine Learning Model for Assistance of Visually Impaired", In Year: 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), pp. 1-6.
8. Aung; Maneetham; Crisnapati; Thwe, Year: 2024, "Enhancing Object Recognition for Visually Impaired Individuals using Computer Vision".
9. Gupta; Chakraborti; Yogitha; Mathivanan, Year: 2022, "Object Detection with Audio Comments using YOLO v3", In 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pp. 903-909.
10. Lin; Chiang; Wu; Yao; Chen Year: 2020, "Smart glasses application system for visually impaired people based on deep learning", In Year: 2020 Indo-Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN) pp. 202-206.
11. Bobby; Praneel; Athuljith; Reerja, Year: 2023. "EBICS: Even Blind I Can See, A Computer Vision Based Guidance System For Visually Impaired", In 2023 International Conference on Control, Communication and Computing (ICCC) pp. 1-6.
12. Desarda K; Gulve S; Hase S; Raut R, Year: 2022, "Single Shot Detector Based Object Detection for Visually Impaired People", In 2022 4th International Conference on Smart Systems and Inventive Technology (ICCSIT), IEEE, pp. 1378-1385.
13. Kumar K; Mappoli P, Year: 2020, "An intelligent Assistant for the Visually Impaired & blind people using machine learning", June 2020 International Journal of Imaging and Robotics, Vol: 20, no: 3.



14. Mahendru M; Dubey S K, Year: 2021, “Real time object detection with audio feedback using Yolo vs. Yolo\_v3”, in 2021 11th international conference on cloud computing, data science & engineering (confluence), IEEE, pp. 734-740.
15. Blue S T; Brindha M, Year: 2019, “Edge detection based boundary box construction algorithm for improving the precision of object detection in YOLOv3”, in 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, pp. 1-5.
16. Ashath Thauth; S, Karishma M, "Artificial Vision - YOLOv8 Algorithm for Objects and Impediments Detection", International Journal of Creative Research Thoughts (IJCRT), ISSN: 2320-2882, Vol: 12, Issue 6, pp.c447-c459.
17. Amjoud A B; Amrouch M, Year: 2023, “Object detection using deep learning”, CNNs and vision transformers: A review. IEEE Access, Vol: 11, pp. 35479-35516.
18. Guravaiah K; Bhavadeesh Y S; Shwejan P; Vardhan A H; Lavanya S, Year: 2023, “Third eye: object recognition and speech generation for visually impaired”, Procedia Computer Science, Vol: 218, pp. 1144-1155.