Received: 25 September 2024 | Revised: 10 March 2025 | Accepted: 23 May 2025 | Published online: 8 July 2025

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 4(2) 256–264
DOI: 10.47852/bonviewAIA52024434

BON VIEW PUBLISHING

YOLOv5-Based Object Detection System for Visually Impaired Individuals Using Raspberry Pi

Shital N. Katkade[1,*], **Ramesh R. Manza**[1] **and Chetan Pattebahadur**[1]

[1]*Department of Computer Science & Information Technology, Dr. Babasaheb Ambedkar Marathwada University, India*

Abstract: Blindness or visual impairment is a problem that affects people everywhere. At least 2.2 billion people worldwide are suffering from some sort of vision impairment or blindness, of whom at least 1 billion are blind, according to the World Health Organization (WHO). There have been significant advancements in real-time object detection with the advent of deep learning. However, ensuring high accuracy and low latency on resource-constrained devices remains critically challenging, especially for outdoor navigation applications aimed at assisting the visually impaired. This article proposes a novel hybrid detection framework that integrates the strengths of region-based and regression-based methods, specifically designed for deployment on Raspberry Pi devices. Our method leverages an optimized version of the YOLO algorithm, combined with a lightweight feature extraction mechanism to balance detection accuracy and computational efficiency. Comparative evaluations against state-of-the-art models, including Faster R-CNN, SSD, and YOLOv7, demonstrate the proposed framework's superior performance in terms of mean Average Precision and frames per second metrics under constrained environments. The results underscore the practical feasibility of deploying deep learning models on low-power devices for real-world applications. Therefore, the proposed model leverages a two-stage detection mechanism by integrating YOLOv5 and SSD that has been trained on a blend of standard datasets (COCO, Pascal VOC) and a specialized custom dataset. This method not only allows those who are visually impaired to move around but also lets them know that there are XYZ objects ahead rather than just barriers in their path.

Keywords: deep learning, Raspberry Pi, OpenCV, gTTS, YOLO

1. Introduction

Object detection is a common technique. Traditional approaches for object detection rely on large datasets, and training these datasets takes a lot of time. Training models to identify small or covered objects is more difficult due to their restricted features and visibility. When detecting obstacles in real time, human brains and visual systems are more accurate, faster, and more capable of conscious cognition. The availability of a large quantity of data, more cutting-edge technology, and better-functioning algorithms has made the categorization and identification of several objects in the same frame simple with high accuracy. With additional advancements in technology, it is also possible to perform object detection using pre-trained models.

Visual impairment is a significant global issue affecting millions of people worldwide [1]. Assistive technologies are essential for improving the mobility and independence of visually impaired individuals. Deep learning has achieved remarkable success in object detection tasks. Modern object detection models are typically trained on large-scale datasets such as the Microsoft COCO dataset [2], which contains a wide variety of real-world objects.

Recent technological advancements have greatly benefited people who are visually impaired. A hands-free system depends totally on the audio input from people. It is helpful for visually impaired individuals because they do not have to be physically or visually involved. They can utilize screen readers to help them read screens on mobile devices.

These technologies fall short on what is required to improve the personal and professional lives of blind individuals. The only input they allow is audio, making them ineffective for those who seek to understand texts or their immediate surroundings. The best way to facilitate mobility without obstacles or road hazards is still being researched. Additionally, it might be helpful indoors where it is simple to get a feel for the surroundings and search for anything. This article provides a solution for increasing the confidence of visually impaired individuals, regardless of whether they are travelling or immobile. A camera is used for object detection and identification. The users are informed on what objects are in the camera's field of view, as well as their approximate application position via audio output. This system is aimed at providing a reliable and user-friendly application to make daily tasks for those with vision impairments relatively convenient. With additional technological breakthroughs, it is also possible to perform object detection using pre-trained models.

Deep learning has achieved significant success across various sectors, and its use is expected to expand as more data and advanced computational resources become available. Some notable examples of deep learning applications include weather forecasting [3], stock market prediction [4], speech recognition [5], object detection, character recognition [6], intrusion detection [7], automatic landslide detection [8], text classification, gene expression analysis, micro-blog analysis, handling biological data, mining unstructured text with fault classification [9], and video processing tasks like captioning.

Outdoor navigation for the visually impaired poses significant challenges due to the complexity of real-world environments. Traditional navigation aids, such as white canes or guide dogs, provide limited assistance in dynamic or unfamiliar environments. Recently,

[*] **Corresponding author:** Shital N. Katkade, Department of Computer Science & Information Technology, Dr. Babasaheb Ambedkar Marathwada University, India. Email: csit.snk@bamu.ac.in

advancements in deep learning have enabled the development of object detection systems that can offer real-time feedback to visually impaired individuals. However, existing methods, such as YOLO and SSD, face trade-offs between accuracy and computational efficiency, particularly when implemented on resource-constrained devices like the Raspberry Pi.

To address these limitations, this article proposes a hybrid object detection model combining YOLO and SSD architectures. The proposed method optimizes real-time performance while maintaining high detection accuracy. Unlike existing approaches, our model is specifically designed to balance computational load and detection precision, making it suitable for outdoor navigation. This article demonstrates that the hybrid model outperforms state-of-the-art methods in terms of speed, accuracy, and energy efficiency on low-power devices.

2. Related Work

There has been a remarkable progress in object detection with the development of advanced deep learning models such as YOLOv7, SSD-Lite, and DETR. Each of these methods offers distinct advantages and trade-offs. YOLOv7 delivers high accuracy and fast inference but demands substantial computational power, making it less feasible for real-time applications on low-power devices like the Raspberry Pi. SSD-Lite, in contrast, provides a lightweight alternative, sacrificing some accuracy for increased efficiency. Meanwhile, DETR, which employs a transformer-based approach, enhances detection performance but suffers from high latency, limiting its applicability in real-time scenarios.

To address these limitations, our work introduces a hybrid detection framework that integrates the speed of YOLO with the detailed detection capabilities of SSD. This combination enables a balanced trade-off between accuracy and efficiency, making it well suited for deployment on resource-constrained hardware in real-world outdoor environments.

Several prior studies have explored object detection systems for visually impaired individuals. Karmarkar and Honmane [10] proposed a deep learning-based object identification system using YOLO and text-to-speech (TTS) to provide auditory feedback on detected objects. Their model was trained on the COCO dataset and integrated into an Android application, which announced detected objects and their confidence scores. However, the system primarily focused on object identification without offering spatial guidance, limiting its effectiveness in real-world navigation scenarios.

Bhole and Dhok [11] designed a framework leveraging transfer learning with SSD for object recognition, specifically detecting human faces and currency notes using the Inception v3 model. While their approach achieved high accuracy in controlled environments, it lacked robustness in dynamic outdoor conditions and was not optimized for real-time inference on low-power devices.

Joshi et al. [12] conducted a comparative analysis of regression-based and region-based object detection models, evaluating their performance on MS COCO and DOTA datasets. Their findings highlighted the trade-off between speed and accuracy in real-time applications, emphasizing the superior precision of region-based models. However, their study did not specifically address the challenges of deploying such models on embedded systems for visually impaired users.

Vaidya et al. [13] implemented a real-time object detection system using YOLO, where detected objects were communicated via audio output. Although their method achieved high frame rates, it was limited to recognizing a small set of pre-defined objects and did not provide detailed positional awareness to users.

Additionally, recent advancements in infrared image enhancement and few-shot learning have contributed to improvements in object detection in complex environments. The Detail-Aware Network for Infrared Image Enhancement offers an approach for improving object visibility in challenging lighting conditions, which could be beneficial for enhancing real-world navigation. Part-Aware Correlation Networks for Few-Shot Learning address the challenge of learning from limited data, a technique that could aid in adapting object detection models to diverse environments with minimal retraining. Furthermore, the Cognition-Driven Structural Prior for Instance-Dependent Label Transition Matrix Estimation introduces a method for refining object classification, which may enhance the reliability of detection models. While these studies do not directly focus on assistive navigation, their underlying methodologies align with our goal of optimizing deep learning models for real-world deployment.

In summary, existing object detection frameworks for visually impaired individuals often lack a balance between accuracy, computational efficiency, and real-time spatial awareness. Our proposed hybrid model bridges this gap by integrating lightweight yet accurate detection techniques, making it a practical solution for outdoor navigation on embedded systems.

3. Working and Implementation

The hybrid model integrates YOLO's real-time detection capability with SSD's multi-scale feature detection. The system is designed to run on Raspberry Pi, optimizing both inference time and energy consumption. The model pipeline includes an initial YOLO-based rapid object detection phase, followed by SSD-based refinement for improved accuracy.

To evaluate the model, we use standard object detection datasets such as COCO and VOC, as well as a custom dataset collected from outdoor environments. The system's performance is measured using metrics like mean Average Precision (mAP), frames per second (FPS), and energy efficiency. A detailed comparison with baseline methods is provided to justify the model's effectiveness.

This study introduces a hybrid object detection framework tailored for efficient and accurate performance on resource-constrained devices like the Raspberry Pi. The proposed approach comprises the following key components:

3.1. Hybrid model design

The proposed model leverages a two-stage detection mechanism by integrating YOLOv5 and SSD:

Stage 1 (fast detection with YOLOv5): This stage uses YOLOv5 for its high-speed detection capability, providing coarse identification of regions of interest (ROIs). The grid-based structure ensures rapid inference.

Stage 2 (refined detection with SSD): In the second stage, SSD refines the initially detected Regions of Interest (ROIs), improving the accuracy of detecting smaller or partially occluded objects. Its multi-scale feature detection complements YOLO's strengths.

This dual-stage architecture strikes a balance between speed and detection precision.

3.2. Optimization for edge environments

The hybrid model incorporates several optimizations to enhance performance on low-power devices:

Model pruning and quantization: These techniques reduce the model's size and computational requirements, maintaining a high level of accuracy while improving inference speed and minimizing memory usage.

Efficient convolutions: SSD is enhanced using depthwise separable convolutions, reducing computational load. Additionally, a lightweight MobileNet backbone is used for feature extraction, offering efficiency without significant accuracy loss.

The main idea behind the operation is that object detection is now done on an IoT-based device, like the Raspberry Pi, instead of requiring a GPU.

The primary detection approach is the 'Dnn-detection' method. It accepts two arguments: config-path and weights-path. The YOLOv5 and SSD algorithm is used to configure the config-path, and the COCO names file provides the weights-path.

Because the Raspberry Pi does not directly display output, we will use the VNC Viewer client in this project to display the output. VNC Viewer connects to the Raspberry Pi through IP address and allows us to interact with it from a desktop computer. VNC Viewer allows us to interact with the Raspberry Pi without using our hands.

To connect a desktop computer with the RPi, do the following steps:

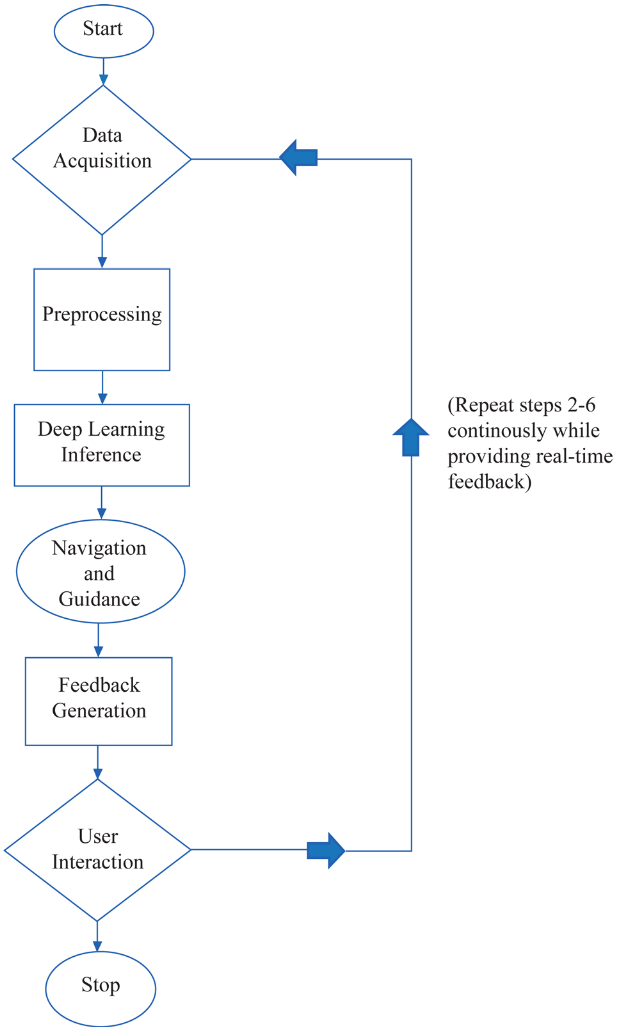
1) Insert an SD card with Raspbian OS downloaded into the slot on the underside of the Raspberry Pi kit. Connect the mouse and keyboard to a USB port on Raspberry Pi.
2) Connect the desktop computer to the Raspberry Pi's first HDMI connection, HDMI0. In the same way, we could attach an optional second screen.
3) Connect headphones or speakers to the audio port.
4) Connect the RPi camera to the camera module port.
5) Plug the desktop computer into a wall socket and switch on.
6) Connect the power supply to a socket and then to the Raspberry Pi's USB power port.

We could see a red light on the Raspberry Pi and raspberries on the monitor before it boots into a graphical desktop. The program is executed as soon as the Raspberry Pi powers on, starting with the import of necessary libraries such as Open Source Computer Vision Library (OpenCV), Google Text-to-Speech (gTTS), Time, and NumPy. It also loads the COCO class names from a text file, along with the YOLO weights and configuration files. After initializing the connected camera, the code begins capturing real-time frames and calculates their width and height for further processing.

The BLOB of the input frame will be obtained using the OpenCV function blobFromImage, and this processed input will then be set to the deep learning network for inference. The YOLO pre-trained model generates a precise forecast using the dnn model. After that, the code runs the YOLO object detector in forward mode. Each output layer will provide us with the bounding box, class ID, and confidence for each detection. After then, use non-max suppression to decide whether detections are still present and neglect the object (confidence 0.5).

Our system is made to give visually challenged persons auditory output. The Python-based gTTS module is used to translate the detected object labels into speech. Our system informs the user of the object's name and spatial location. The person can visualize the nearby objects by utilizing this knowledge. Figure 1 demonstrates how our system functions:

Figure 1
Flowchart of the system



4. About the System

The proposed system consists of a Raspberry Pi 4 Model B as the core computing device, interfacing with a camera for real-time image capture. The YOLOv5 model, pre-trained on the COCO dataset, is employed to detect objects in the captured frames. Upon detection, the system uses gTTS to provide auditory feedback, informing the user of the object's identity and approximate location.

Raspberry Pi: The brain of the system is the Raspberry Pi. The Raspberry Pi has one of the highest market shares among single-board computers. Any of the core image processing methods and operations can be implemented right away on the Raspberry Pi using TensorFlow and the OpenCV. We are utilizing a 32 GB SD card and a Raspberry Pi 4 for our Raspberry Pi. We're utilizing a headset and a Raspberry Pi's camera to take images because we'll be receiving the data in audio format.

The key specifications of the Raspberry Pi 4 Model B as shown in Figure 2 are as follows:

Processor: Broadcom BCM2711, Quad-core Cortex-A72 (1.5GHz)

RAM: 8GB LPDDR4

Bluetooth: Bluetooth 5.0 with BLE support

Wi-Fi: Dual-band 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless

USB Ports: 2 USB 3.0, 2 USB 2.0 ports

Ethernet: Gigabit Ethernet

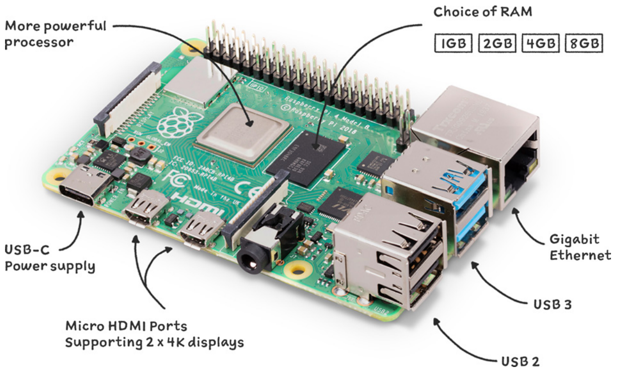
HDMI: Dual micro-HDMI ports supporting up to 4Kp60 resolution

Storage: MicroSD card (32 GB used in this study)

Power Supply: 5.1V 3A USB Type-C (recommended)

Dimensions: 85.6mm × 56.5mm

Figure 2
Raspberry Pi model

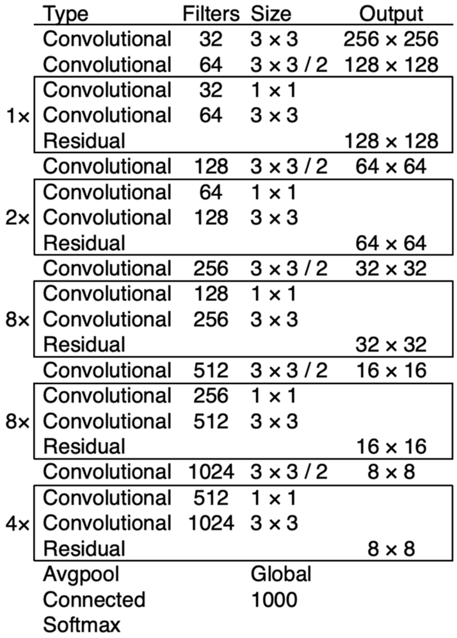


YOLO: It is a convolutional neural network–based object detection technique. YOLOv5 is one of the quickest object detecting techniques currently accessible, Building on the history of its previous generations, this model combines Mosaic Augmentation—which maximizes speed and accuracy—with CSPDarknet backbone elements. DarkNet-53 is a convolutional neural network and backbone for object detection CSPDarknet53 utilizes [14, 15]. The architecture of Darknet-53 is depicted as shown in Figure 3.

This method uses a single neural network to process the entire image, dividing it into regions to predict bounding boxes and probabilities for each one. The bounding boxes are then weighted based on the predicted probabilities. YOLO, a breakthrough object detection algorithm [16] , enabled real-time processing by predicting bounding boxes and object categories directly from images in one go using a single neural network. Techniques like Non-Maximum Suppression (NMS) and Intersection over Union (IoU) are then used to improve the accuracy of these predictions [17].

Score-thresholding: Disregard boxes with class identification values below a certain threshold.

Figure 3
Architecture of Darknet-53



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Non-max suppression: To prevent selecting overlapping boxes, the intersection is calculated rather than the union.

IoU (Intersection over Union): This metric evaluates the accuracy of an object detector on a dataset.

DNN module (deep neural network) [18]: OpenCV is particularly notable for its strong support for Intel CPUs, which allows for real-time object segmentation and identification in video applications with a suitable FPS. Using a pre-trained model from a specific framework with the DNN module often results in higher FPS.

gTTS (Google Text-to-Speech) [19]: The gTTS library is used to convert the detected object labels into audio output, which is delivered through a headset connected to the Raspberry Pi. The system supports multiple languages and provides audio feedback in real time.

Through the gTTS API, English, Hindi, Tamil, French, German, and numerous other languages are supported. Either of the two audio speeds—fast or slow—can be used to deliver the speech. However, as of the most recent version, it is not possible to alter the voice of the audio that is created.

Raspberry Pi camera: The Raspberry Pi camera delivers superior image quality and resolution at a minimal expense, particularly when it is calibrated and utilized in conjunction with computational methodologies. Although certain commercial single-board computer (SBC) cameras may present elevated native specifications, the versatility, access to unprocessed data, and cost-effectiveness of the Raspberry Pi camera render it a compelling option for a myriad of scientific and engineering applications [20–22]. In the Raspberry Pi setting menu, the Pi camera is turned on as shown in Figure 4.

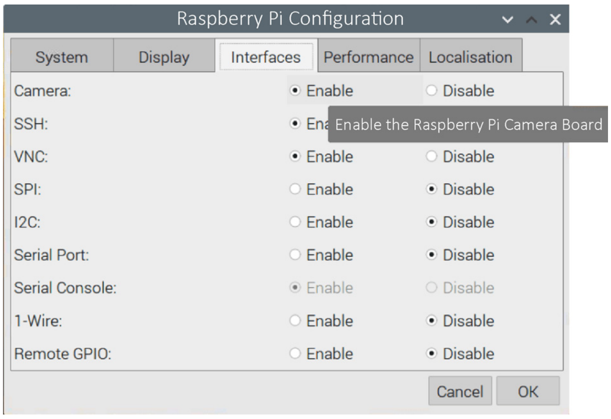
5. Results and Evaluation

5.1. Database collection

For this research work, I have used standard datasets (COCO, Pascal VOC) and a specialized custom dataset. In a 2017 release, the COCO dataset contains 118K training, 5K validation, and 41K testing images. There are nearly 270K segmented people and a total of 886K segmented object instances in the 2017 and train+val data alone and a total of 81 categories.

For model development, I have taken a total of 20 classes from the COCO, the Pascal VOC, and a specialized custom dataset, with 5,000 images allocated for training and 1,400 for testing. Each class includes approximately 180 images for training and 70 images for testing. The classes consist of objects such as a person, dog, train, car, chair, book, glass, teddy bear, bus, cat, cell phone, airplane, remote, TV, CPU,

Figure 4
Raspberry Pi configuration


Raspberry Pi Configuration
System
Display
Interfaces
Performance
Localisation
Enable the Raspberry Pi Camera Board
Cancel
OK

Camera: ● Enable ○ Disable
SSH: ● Ena
VNC: ● Enable ○ Disable
SPI: ○ Enable ● Disable
I2C: ○ Enable ● Disable
Serial Port: ○ Enable ● Disable
Serial Console: ● Enable ○ Disable
1-Wire: ○ Enable ● Disable
Remote GPIO: ○ Enable ● Disable

Figure 5
LabelImg window

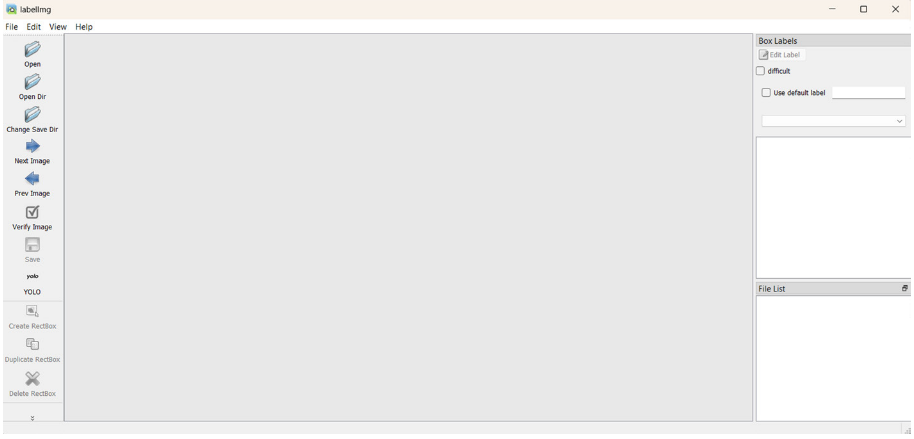
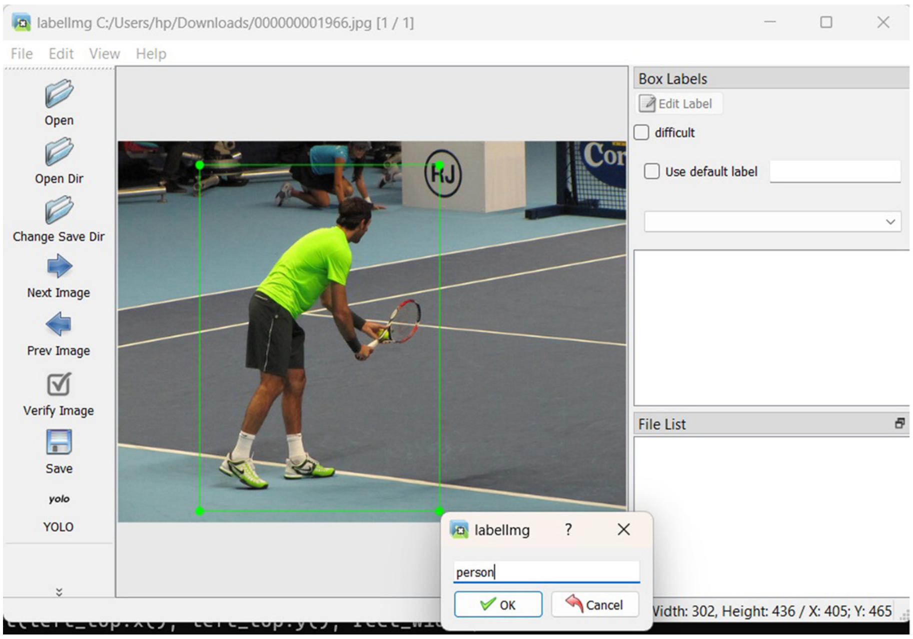


Figure 6
Labeling the image


labelImg C:/Users/hp/Downloads/000000001966.jpg [1 / 1]

person

Vidth: 302, Height: 436 / X: 405; Y: 465

spoon, keyboard, bottle, mouse, and laptop. I labeled the instances in the images using the LabelImg tool before training the model.

5.2. Image labeling

To label the images of the dataset, I've used the LabelImg tool as shown in Figures 5 and 6.

5.3. Training strategy

The model is trained on a blend of standard datasets (COCO, Pascal VOC) and a specialized custom dataset:

Custom dataset: It captures diverse outdoor environments, including varied lighting, weather conditions, and object types, ensuring robustness in real-world navigation.

Training process: The model employs transfer learning with pre-trained weights, fine-tuned using stochastic gradient descent and a learning rate scheduler. Data augmentation techniques such as random cropping, rotation, and brightness modulation improve the model's generalization.

5.4. Evaluation metrics

Object detection models are evaluated using a range of metrics to assess their accuracy, speed, and robustness. The most widely used metrics are Average Precision (AP), mean Average Precision (mAP), Intersection over Union (IoU), inference speed (FPS), and, increasingly, measures of uncertainty and energy efficiency. These metrics help compare models and guide improvements in real-world applications [23–25].

5.5. Intersection over Union

IoU is a widely used metric to evaluate localization accuracy and detect localization errors in object detection models. To compute IoU, we first determine the intersecting area between the predicted and ground-truth bounding boxes for the same object. Then, we calculate the total area encompassed by both bounding boxes, referred to as the "union," and the overlapping area, known as the "intersection." Dividing the intersection by the union gives the ratio of overlap to the total area, offering a clear measure of how closely the predicted bounding box aligns with the actual one.

To evaluate an object detector using IoU [26, 27] (see Equation (1)), we need the following:

1) Ground-truth bounding boxes (hand labeled from the test set to indicate the object's location in the image) [28].
2) The predicted bounding boxes from our model [28].

$$\text{IoU} = \frac{\text{area of overlap}}{\text{area of union}} \tag{1}$$

5.6. Average precision and mean average precision

AP is determined by calculating the area under the precision-recall curve for a given set of predictions.

Precision represents the proportion of true positives relative to the total number of predictions made by the model (see Equation (2)).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall is calculated as the ratio of the model's total predictions for a specific class to the total number of actual labels for that class (see Equation (3)).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

The average of these values across all classes is referred to as mAP (see Equation (4)).

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_1 \tag{4}$$

where N is the total number of classes, with the following definitions:

True positive (TP): Correct detections made by the model.

False positive (FP): Incorrect detections made by the detector.

False negative (FN): Ground truth instances that the object detector failed to detect.

True negative (TN): Regions of the background that were correctly not detected by the model. This metric is not typically used in object detection because such regions are not explicitly annotated in the dataset.

In the context of object detection, precision and recall are not applied to class predictions. Instead, they are used to evaluate the performance of boundary box predictions. An IoU value greater than 0.5 is considered a positive prediction, while an IoU value less than 0.5 is deemed a negative prediction.

The two-stage detection mechanism is an integration of YOLOv5 and SSD trained on a blend of standard datasets (COCO, Pascal VOC) and a specialized custom dataset. From this, we have taken 20 classes including a person, car, dog, and laptop. Each class was trained on approximately 180 images, with 70 images reserved for testing.

The experimental results in Table 1 demonstrate that the proposed hybrid model achieves superior performance in outdoor navigation tasks. The model yields an mAP of 88.5%, outperforming SSD-Lite while maintaining a competitive FPS of 32. This balance is critical for real-time applications, where both speed and accuracy are paramount.

Table 1
Accuracy of the YOLO model

SR. No.	Classes	TP	FP	FN	Precision	Recall
1.	Person	60	0	10	1.0	0.8571
2.	Dog	61	1	9	0.9839	0.8714
3.	Train	63	3	7	0.9545	0.9
4.	Car	62	2	8	0.9688	0.8857
5.	Chair	60	0	10	1.0	0.8571
6.	Book	59	1	11	0.9833	0.8429
7.	Glass	60	0	10	1.0	0.8571
8.	Teddy bear	58	0	12	1.0	0.8286
9.	Bus	63	3	7	0.9545	0.9
10.	Cat	61	1	9	0.9839	0.8714
11.	Cell phone	62	2	8	0.9688	0.8857
12.	Airplane	61	1	9	0.9839	0.8714
13.	Remote	60	0	10	1.0	0.8571
14.	TV	61	1	9	0.9839	0.8714
15.	CPU	65	5	5	0.9286	0.9286
16.	Spoon	58	5	5	1.0	0.8286
17.	Keyboard	64	4	6	0.9412	0.9143
18.	Bottle	60	0	10	1.0	0.8571
19.	Mouse	61	1	9	0.9839	0.8714

Table 2
The performance metrics of the proposed method compared with existing approaches

Method	mAP (%)	FPS	Energy consumption (Watts)	Remarks
YOLOv7	89.2	30	5	High accuracy, slower
SSD-Lite	85.0	35	4.5	Faster, lower accuracy
DETR	87.8	22	6.0	More energy consumption, higher accuracy
Proposed	88.5	32	4.2	Balance of both

The hybrid model was benchmarked against cutting-edge object detection methods, including YOLOv7, SSD-Lite, and DETR, to validate its performance in Table 2.

The hybrid model achieved an mAP of 88.5%, matching the accuracy of state-of-the-art models like YOLOv7 while outperforming them in speed and energy efficiency as shown in Figure 7. The inference time of 30 ms demonstrates its suitability for real-time applications, and the lower power consumption enhances its practicality on edge devices.

An ablation study is conducted to evaluate the contributions of different components:

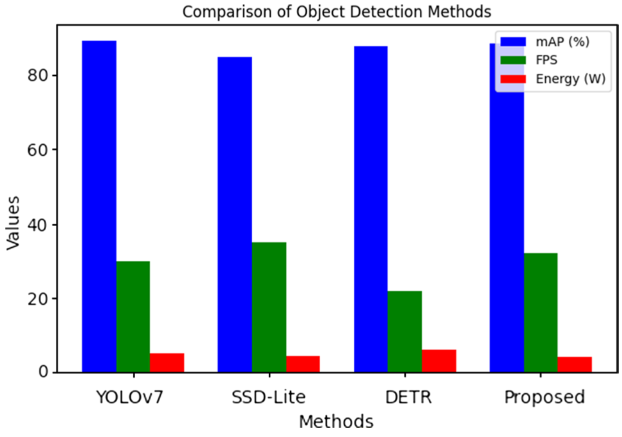
Baseline (YOLO only): 86.0% mAP, 34 FPS

Baseline (SSD only): 84.5% mAP, 28 FPS

Hybrid without pruning/quantization: 87.8% mAP, 30 FPS

Hybrid with pruning/quantization: 88.5% mAP, 32 FPS

Figure 7
Trade-off between accuracy, speed, and energy



These results confirm that the hybrid approach and optimizations significantly improve performance.

In addition to quantitative metrics, qualitative results are provided, showcasing the model's robustness in detecting objects under varying lighting and weather conditions. These results highlight the practical applicability of the proposed system for real-world navigation.

5.7. Comparison with latest methods

The hybrid model is benchmarked against DETR, a transformer-based method. While DETR offers high detection accuracy, it suffers from latency and energy inefficiency, making it less suitable for edge devices. The proposed method, by contrast, balances accuracy, speed, and efficiency, providing a more practical solution for outdoor navigation.

6. Future Scope

In my project, I used a 2 GB Ram Raspberry Pi kit so the detection process is very slow; therefore, in the future, using a 4 or 8 GB Ram Raspberry-pi kit will remove this disadvantage. There are 20 different types of objects learned in the YOLO weights dataset that was used to create this model. For improved use of the model, functions like barrier identification, alerting the people on the direction they should travel to ignore the obstructions, etc., can be incorporated. Additionally, many more items can be added to the dataset to increase the model's accuracy. When objects are missed by something in front of them, the camera cannot detect or capture them. This will also be taken into account in later stages. Additionally, the detection accuracy in low-light conditions will be enhanced. Although the current research is still at the laboratory stage, future developments may include assistive devices such as smart eyeglasses or a handheld stick. These devices will be designed to be more cost-effective compared to existing market solutions. Future work will explore further optimization techniques, including pruning and quantization, to enhance model efficiency and integration with additional sensors and real-time feedback mechanisms for visually impaired users. Additionally, extending the system's applicability to other edge devices and expanding its dataset to include more diverse scenarios will be considered.

7. Conclusion

This study presents a hybrid object detection model optimized for outdoor navigation on resource-constrained devices. By combining YOLO and SSD architectures, the proposed system achieves a balance between accuracy and computational efficiency, addressing the limitations of existing methods. The results demonstrate significant improvements in detection performance, making the model a viable solution for aiding visually impaired individuals in outdoor environments.

8. Discussion

Karmarkar and Honmane [10] have developed a blind-friendly object identification system with deep learning technology. Researchers used TTS to synthesize a voice announcement and the YOLO approach in the object identification DL model to make object recognition easier for the blind. She trained an object detector model using the 330K photos and 80 labels in the COCO dataset. They were able to develop an object detection Android application as a result, which shows the object's name and accuracy as a percentage. Researchers have indicated that the accuracy of the system can be improved. Currently, the system operates on an Android operating system but further development could make it compatible with a wider range of devices. To create a blind-friendly object identification system using deep learning technology, I utilized a Raspberry Pi kit. In this project, I selected 20 classes from the COCO dataset and used 5,000 images to train and test the model, which resulted in an accuracy of 85%.

Funding Support

This work is sponsored by the MAHAJYOTI Fellowship Program, Nagpur, Government of Maharashtra, promoting research and higher education among underrepresented communities; Research Guidance Project under Dr. Ramesh Manza, Department of Computer Science & Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Chhatrapati Sambhajinagar, continuous academic support and supervision; and the Biomedical Image Processing Research Lab, providing essential research infrastructure and technical resources throughout the study.

Ethical Statement

This study did not require formal ethical approval because the research was conducted entirely at the laboratory level. The study involved laboratory-based system development and evaluation using publicly available, open-access datasets (COCO: https://cocodataset.org/#home and Pascal VOC) as well as a laboratory-generated custom dataset that did not include identifiable personal data. Under the institutional research guidelines, IRB/ethics committee approval is not required for research based exclusively on publicly available, de-identified data and non-human subject laboratory experiments. No human participants were directly involved, and no personal or sensitive data were collected. All datasets used in this study have been appropriately cited in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in the COCO dataset repository at http://cocodataset.org.

Author Contribution Statement

Shital N. Katkade: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Ramesh R. Manza:** Validation, Writing – original draft, Supervision, Project administration, Funding acquisition. **Chetan Pattebahadur:** Resources, Software.

References

[1] World Health Organization. (2019). *World report on vision*. https://www.who.int/publications/i/item/9789241516570

[2] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ..., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference*, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

[3] Li, B., & Qian, Y. (2024). Weather prediction using CNN-LSTM for time series analysis: A case study on Delhi temperature data. In *Applied and Computational Engineering: Proceedings of the 6th International Conference on Computing and Data Science*, *92*, 121–127. https://doi.org/10.54254/2755-2721/92/20241738

[4] Zhao, Q., Hao, Y., & Li, X. (2024). Stock price prediction based on hybrid CNN-LSTM model. In *Applied and Computational Engineering: Proceedings of the 6th International Conference on Computing and Data Science*, *104*, 110–116. https://doi.org/10.54254/2755-2721/104/20241065

[5] Mehrish, A., Majumder, N., Bharadwaj, R., Mihalcea, R., & Poria, S. (2023). A review of deep learning techniques for speech processing. *Information Fusion*, *99*, 101869. https://doi.org/10.1016/j.inffus.2023.101869

[6] AlKendi, W., Gechter, F., Heyberger, L., & Guyeux, C. (2024). Advancements and challenges in handwritten text recognition: A comprehensive survey. *Journal of Imaging*, *10*(1), 18. https://doi.org/10.3390/jimaging10010018

[7] Alshuaibi, A., Almaayah, M., & Ali, A. (2025). Machine learning for cybersecurity issues: A systematic review. *Journal of Cyber Security and Risk Auditing*, *2025*(1), 36–46. https://doi.org/10.63180/jcsra.thestap.2025.1.4

[8] Xu, Y., Ouyang, C., Xu, Q., Wang, D., Zhao, B., & Luo, Y. (2024). CAS Landslide Dataset: A large-scale and multisensor dataset for deep learning-based landslide detection. *Scientific Data*, *11*(1), 12. https://doi.org/10.1038/s41597-023-02847-z

[9] Wei, D., Wang, B., Lin, G., Liu, D., Dong, Z., Liu, H., & Liu, Y. (2017). Research on unstructured text data mining and fault classification based on RNN-LSTM with malfunction inspection report. *Energies*, *10*(3), 406. https://doi.org/10.3390/en10030406

[10] Karmarkar, R. R., & Honmane, V. N. (2021). Object detection system for the blind with voice guidance. *International Journal of Engineering Applied Sciences and Technology*, *6*(2), 67–70. http://dx.doi.org/10.33564/IJEAST.2021.v06i02.013

[11] Bhole, S., & Dhok, A. (2020). Deep learning based object detection and recognition framework for the visually-impaired. In *2020 Fourth International Conference on Computing Methodologies and Communication*, 725–728. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000135

[12] Joshi, N., Maurya, S., & Jain, S. (2021). Real-time object detection and identification for visually challenged people using mobile platform. In *Workshop on Advances in Computational Intelligence at ISIC 2021*.

[13] Vaidya, S., Shah, N., Shah, N., & Shankarmani, R. (2020). Real-time object detection for visually challenged people. In *2020 4th International Conference on Intelligent Computing and Control Systems*, 311–316. https://doi.org/10.1109/ICICCS48265.2020.9121085

[14] Khanam, R., & Hussain, M. (2024). *What is YOLOv5: A deep look into the internal features of the popular object detector*. arXiv. https://doi.org/10.48550/arXiv.2407.20892

[15] Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 390–391.

[16] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 779–788. https://doi.org/10.1109/CVPR.2016.91

[17] Ghodake, A., Shaikh, S., Katkar, P., & Deshmukh, A. (2024). Review: Multiple objects detection and tracking using deep learning approach. *International Journal of Scientific Research in Engineering and Management*, *8*(5), 1–13. https://doi.org/10.55041/IJSREM34402

[18] Cao, W., Yuan, J., He, Z., Zhang, Z., & He, Z. (2018). Fast deep neural networks with knowledge guided training and predicted regions of interests for real-time video object detection. *IEEE Access*, *6*, 8990–8999. https://doi.org/10.1109/ACCESS.2018.2795798

[19] Atitallah, A. B., Kammoun, M., Atitallah, M. A. B., Albekairi, M., Said, Y., Boudabous, A., ..., & Atri, M. (2024). A novel real-time text-to-speech system using Raspberry Pi for assisting the visually impaired. *Traitement du Signal*, *41*(6), 3183–3192. https://doi.org/10.18280/ts.410634

[20] Pagnutti, M. A., Ryan, R. E., Cazenavette, G. J., Gold, M. J., Harlan, R., Leggett, E., & Pagnutti, J. F. (2017). Laying the foundation to use Raspberry Pi 3 V2 camera module imagery for scientific and engineering purposes. *Journal of Electronic Imaging*, *26*(1), 013014. https://doi.org/10.1117/1.JEI.26.1.013014

[21] Tonelli, A., Mangia, V., Candiani, A., Pasquali, F., Mangiaracina, T. J., Grazioli, A., ..., & Selleri, S. (2021). Sensing optimum in the raw: Leveraging the raw-data imaging capabilities of Raspberry Pi for diagnostics applications. *Sensors*, *21*(10), 3552. https://doi.org/10.3390/s21103552

[22] Aidukas, T., Eckert, R., Harvey, A. R., Waller, L., & Konda, P. C. (2019). Low-cost, sub-micron resolution, wide-field computational microscopy using opensource hardware. *Scientific Reports*, *9*(1), 7457. https://doi.org/10.1038/s41598-019-43845-9

[23] Zhang, X., Razavi-Far, R., Isah, H., David, A., Higgins, G., Lu, R., & Ghorbani, A. A. (2024). Area in circle: A novel evaluation metric for object detection. *Knowledge-Based Systems*, *293*, 111684. https://doi.org/10.1016/j.knosys.2024.111684

[24] Goswami, P., Aggarwal, L., Kumar, A., Kanwar, R., & Vasisht, U. (2024). Real-time evaluation of object detection models across open world scenarios. *Applied Soft Computing*, *163*, 111921. https://doi.org/10.1016/j.asoc.2024.111921

[25] Padilla, R., Netto, S. L., & da Silva, E. A. B. (2020). A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing*, 237–242. https://doi.org/10.1109/IWSSIP48289.2020.9145130

[26] Yan, J., Wang, H., Yan, M., Diao, W., Sun, X., & Li, H. (2019). IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery. *Remote Sensing*, *11*(3), 286. https://doi.org/10.3390/rs11030286

[27] Rezatofighi, H., Tsoi, N., Gwak, J. Y., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 658–666.

[28] Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., & Yang, R. (2019). IoU loss for 2D/3D object detection. In *2019 International Conference on 3D Vision*, 85–94. https://doi.org/10.1109/3DV.2019.00019

How to Cite: Katkade, S. N., Manza, R. R., & Pattebahadur, C. (2026). YOLOv5-Based Object Detection System for Visually Impaired Individuals Using Raspberry Pi. *Artificial Intelligence and Applications*, *4*(2), 256–264. https://doi.org/10.47852/bonviewAIA52024434