

Article

Vision-Based Smart Wearable Assistive Navigation System Using Deep Learning for Visually Impaired People

Syed Salman Shah ¹, Abid Imran ^{1,*} , Saad-Ur-Rehman ¹, Arsalan Arif ¹ , Khurram Khan ², Muhammad Arsalan ¹, Sajjad Manzoor ³ and Ghulam Jawad Sirewal ⁴ 

¹ Faculty of Mechanical Engineering, GIK Institute of Engineering Sciences and Technology, Swabi 23640, Pakistan

² Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Swabi 23640, Pakistan

³ Department of Electrical Engineering, Mirpur University of Science and Technology, Mirpur AJK 10250, Pakistan

⁴ Department of Electrical Engineering Technology, Benazir Bhutto Shaheed University of Technology and Skill Development, Khairpur Mirs 66020, Pakistan; jawadsirewal1@gmail.com

* Correspondence: abid.imran@giki.edu.pk

Abstract

People affected by vision impairment experience significant challenges in mobility and daily life activities. In this paper, a smart assistive navigation system is proposed to address mobility challenges and to enhance the independence of visually impaired individuals. Three modules are integrated into the proposed system. The vision module detects obstacles and interactive objects such as doors, chairs, people, fire extinguishers, etc. The depth cam-based distance module provides the distance of detected objects and obstacles. The voice module provides auditory feedback to visually impaired individuals about the detected objects and obstacles that fall under the pre-defined threshold distance. Finally, the proposed system is optimized in terms of performance and user experience. Jetson Nano is used to reduce the cost of the overall system; however, it has compatibility issues with many of the latest object detection models. The YOLOv5n model is used considering compatibility for object detection, but it has low Mean Average Precision (mAP) and frame rate. To improve the performance of the vision module, various hyperparameters of YOLOv5n are fine-tuned along with transfer learning to enhance the mAP@50 from the original 0.457 to 0.845 and mAP@50-95 from 0.28 to 0.593. Tensor-RT optimization is employed to increase the frame rate to deploy the model in a real scenario. The real-time experimentation shows that the proposed system successfully alerts users to key objects, hazards, and obstacles which enables independent and confident navigation.

Keywords: assistive navigation system; deep learning; visually impaired; Tensor-RT; hyperparameter; YOLO



Academic Editor: Farrokh Janabi-Sharifi

Received: 12 December 2025

Revised: 11 February 2026

Accepted: 26 February 2026

Published: 1 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Visual impairment is experienced by approximately 2.2 billion individuals worldwide. According to projections, there will be 385 million visually impaired individuals worldwide by 2030 out of which 55 million will be blind [1]. However, poverty reduction and improved eye care services helped lower the rate from 4.58% in 1990 to 3.38% in 2015 [2]. Numerous challenges are faced by visually impaired people [3], including limited access to information, lack of accessible printed materials, dependency on mobility aids, difficulties with mobility and spatial awareness, social exclusion, and barriers to employment and healthcare [4,5].

Several long-range, medium-range, and short-range technologies are developed to overcome these challenges [6,7]. Predetermined routes marked by beacons are followed using GPS tracking and landmark databases, which are used by long-range assistive tools to guide users. Audio cues or haptic feedback are also used to facilitate navigation. These solutions are predominantly suited for outdoor wayfinding and lack the ability to detect transient hazards [8]. Information about a user's immediate surroundings can be provided by medium-range tools. These include a white cane and guide dogs as well as sonar or infrared technologies to detect obstacles at greater distances [9]. Electronic travel aids (ETAs) are also introduced to identify obstructions in the user's direct path [10]. However, the reflections can be blocked or changed when people move between the device and the object, or when doors are opened, which affects their performance [11]. Tasks such as reading, object manipulation, and interpersonal communication are performed by short-range tools. Screen layouts and components are presented through speech or braille using screen readers, making computers easier to use for people with visual impairments. Other tools include magnifying text, screens, or scenes optically. Collectively, these tools help overcome barriers related to close-vision tasks [12]. Short-range tools do not provide contextual information about the environment [13]. These technologies enhance the independence of visually impaired individuals, but their limitations highlight the need for the development of more advanced real-time technologies that can offer greater accuracy and faster processing for practical everyday use. Computer vision and deep-learning-based systems are utilized to assist Visually Impaired People (VIP) in both indoor and outdoor environments by not only focusing on obstacle avoidance but also on identifying specific objects that individuals can interact with or avoid based on their nature [14]. Environmental conditions like temperature and surface reflectivity cannot affect this technology, as opposed to traditional methods of using sensors like Lidar, IR, and ultrasonic sensors; however, challenges associated with these systems involve low detection accuracy, poor distance estimation, and delayed feedback. To overcome these challenges, a novel assistive navigation device is proposed using deep learning that detects objects, measures their distances, and provides real-time voice feedback through optimized vision, distance, and voice modules. The novelty of the proposed system lies in its deployment-oriented system optimization and real-time validation on a low-cost embedded platform. The main contributions of this research work are as follows:

- Integrating a distance module that accurately calculates the proximity of detected objects or obstacles to the user.
- Integrating a voice module that conveys essential information about detected objects and obstacles to visually impaired individuals.
- Hyperparameter tuning, optimization, testing numerous combinations, and selecting the optimal combination to enhance the accuracy and speed of computer vision models.
- Achieving seamless integration of three key modules (computer vision module, distance module, and voice module) into a unified intelligent system aimed at assisting visually impaired individuals by reducing real-time latency.

The paper is organized as follows: Section 2 presents the related work; Section 3 details the proposed system methodology and experimental setup; Section 4 covers model training, hyperparameter tuning, and performance evaluation; Section 5 covers real-time testing and comparative analysis; and Section 6 concludes the work.

2. Related Work

Navigating the world is crucial for visually impaired individuals. Assistive technologies have been made more important for navigation due to technological progress, with well-known examples including Way-finder and Envision [15]. Mobile phones have become

a common platform for assistive technologies [16]. Several research activities have been carried out on navigation tools for visually impaired individuals [17]. A comprehensive review is performed discussing different Electronic Travel Aids (ETA) designed for navigation, including machine vision technologies [10].

Traditional assistive methods have been improved through technology, leading to more advanced solutions for people with visual impairments. Comprehensive spatial information is provided by Electronic Orientation Aids (EOAs) to help users know their location through detailed environmental maps, while obstacles and hazards in the immediate path are detected by Electronic Travel Aids (ETAs) to improve safety and movement. To capture broad environmental data, EOAs often integrate cameras and sensors but they may face challenges related to device complexity and weight [18]. In contrast, ETAs employ a range of sensing technologies including general cameras, depth cameras, RFID, Bluetooth beacons, LiDAR, Infrared (IR) sensors, and ultrasonic sensors to identify obstacles and ensure safe navigation [10]. Despite the advancements offered by these technologies, challenges such as device complexity, cost, and limitations in accuracy underscore the need for further refinement and innovation.

Earphones or speakers are commonly used to provide audio feedback in navigation aids. However, when too many auditory cues are given, users can become confused and important sounds from the environment may be missed [19]. This issue is addressed by bone-conduction headphones, as sound is sent directly to the inner ear without blocking the ear canal.

Computer vision algorithms and optical sensors are used by vision-based navigation to detect obstacles and guide users [20–23]. In systems like Tyflos, stereo cameras are used to capture 3D images of the environment, which are then turned into spoken instructions [24]. A single camera for object localization is used by Visual Simultaneous Localization and Mapping (VSLAM) which reduces the need for additional sensors [25]. RGB-D cameras integrate depth sensing for enhanced obstacle detection [26]. However, these vision-based approaches often face challenges such as high computational requirements, sensitivity to lighting conditions, and limited performance in terms of accuracy in various environments. This gap highlights the need to explore alternative methods using IoT (Internet of Things), and non-visual data should also be reviewed to overcome the current limitations and support future integration.

Communication between systems by the Internet of Things (IoT) was proposed which helps navigation systems work more effectively without human involvement. To detect objects and provide navigation alerts, smart canes and IoT technology are used by devices like Indriya and Blind guide [27]. However, these systems often require internet access for full functionality. Navigation systems can also utilize non-visual data, such as ultrasonic sensors, beacons, and infrared detectors, for feature extraction. Proximity is detected using ultrasonic sensors, while GPS is combined with fixed devices for beacon-based localization. Infrared sensors offer contactless distance measurements, providing an alternative to vision-based navigation [24]. Tactile tools like raised maps and magnet boards assist visually impaired individuals in spatial learning. Updated and detailed navigation information was introduced through interactive and augmented reality maps like SmartTactMaps and LucentMaps [28]. These tools combine tactile, audio, and visual feedback for effective navigation assistance. Audio feedback and sonification are used by systems like the Sound of Vision and SVETA to support navigation. Despite these advancements, many IoT- and non-visual-based systems face challenges such as dependency on external infrastructure, limited adaptability, and the need for user training to interpret feedback correctly. Training is needed by users to understand the sounds correctly [29].

Despite technological advancements, navigation assistive devices for visually impaired individuals have low user acceptance due to various shortcomings like the need for better object detection algorithms, proper datasets for indoor scenes, effective feedback mechanisms, and optimized techniques to reduce computational costs. Most existing work has primarily focused on obstacle avoidance, but it is equally important to identify specific objects that visually impaired individuals can interact with or avoid based on their nature. To address these issues, a smart Assistive Navigation system has been proposed to enhance usability and efficiency in indoor environments.

3. Proposed System Architecture

The overall architecture of the proposed system, as illustrated in Figure 1, represents a smart assistive navigation system which detects various interactive objects and finds their distance while notifying the user about them. The overall system contains three modules: the computer vision module, distance module, and voice module, which are all integrated. The computer vision module is responsible for scanning the environment and detecting all the interactive objects in the scene. Once the interactive objects are detected, the distance module finds their distance from the user. Whenever the visually impaired individual gets close to any of the interactive objects or obstacles and crosses some threshold, then the voice module is activated to inform the user about the interactive objects or obstacles and their distances from the user. In this way, all the three modules are integrated to form a complete assistive navigation system used in indoor environments.

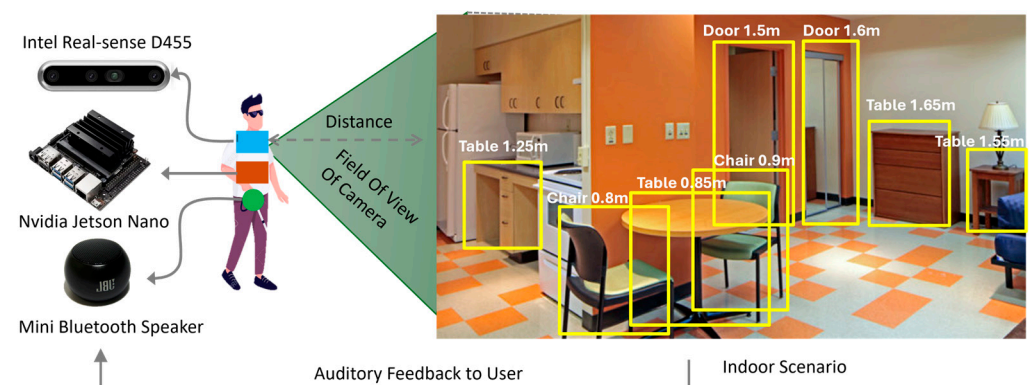


Figure 1. Schematic diagram of overall smart assistive navigation system.

The challenges encountered during the development of the portable and cost-effective assistive navigation system and their proposed optimal solutions are illustrated in Figure 2. To ensure portability and cost effectiveness, the NVIDIA Jetson Nano is selected as the processing platform. However, compatibility issues arise with advanced object detection models, leading to the adoption of YOLOv5, which is well-suited for Jetson Nano. Initially, the model exhibited low detection accuracy and limited processing speed, hindering real-time performance. The accuracy issue is addressed through extensive hyperparameter tuning on our custom dataset and by employing the lightweight YOLOv5n version. Additionally, TensorRT optimization is employed to enhance the inference speed for real-time navigation. Finally, the system was further refined by optimizing the auditory feedback frequency and nature to ensure a user-friendly and efficient experience for visually impaired individuals. The working of each module is explained in detail in the following sub-sections.

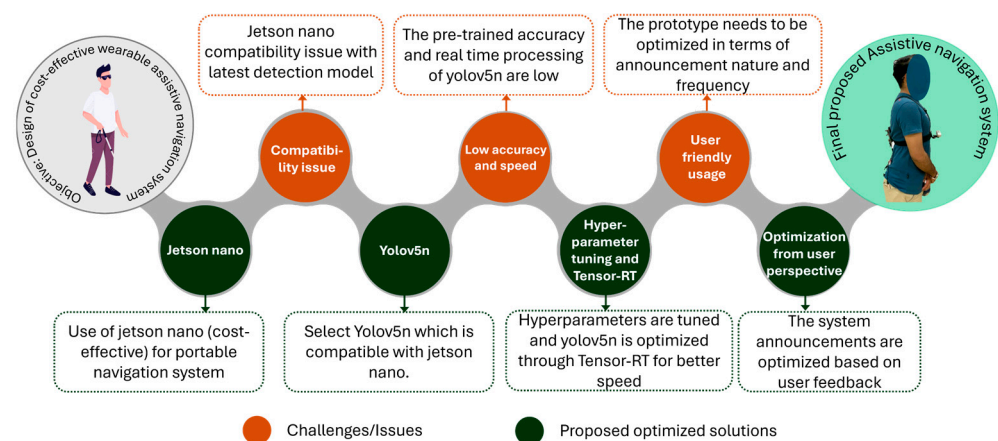


Figure 2. Challenges/issues and their proposed optimized solutions in the development of the navigation system.

3.1. Computer Vision Module

The computer vision module is responsible for detecting objects in real time with a high enough frame rate (15 to 20 frames per second are considered for human walking). This module is further categorized into data acquisition and transfer learning-based YOLOv5n training for real-time detection of various interactive objects in the indoor environment. The following factors are mainly considered to choose and train the computer vision modules: detection accuracy (mAP values), Frame Per Second (FPS), and computational cost.

3.1.1. Data Acquisition

To train and evaluate the proposed object detection system, a new indoor dataset is created. The dataset consists of five classes of objects routinely found within residential environments which are doors, chairs, humans, fires, and fire extinguishers. These five classes are interactive objects while all other objects are considered as obstacles. All the images are taken using the camera of an iPhone XS device under various conditions with a resolution of 1280×720 pixels. Images show objects from different positions and orientations, with complex natural backgrounds, and arbitrary placement relative to one another. Lighting also varied realistically between shots to simulate diverse indoor settings.

The collected dataset having 700 images is manually annotated using Robo-flow, with bounding boxes drawn around each object to provide class labels. To enhance the size and complexity of the dataset, several data augmentation techniques are applied using Robo-flow: Resize, Crop, Rotation, Hue, Exposure, Blur, Noise, Cutout, and Auto-Orient. Hence the dataset is enhanced to 1000 images (200 images per class). Some images taken from our final dataset are given in Figure 3. The data are then split into three sets including the training set which contains 70% of the images (700 images), the validation set containing 15% of the images (150 images), and the test set containing 15% of the images (150 images). This setup ensured an unbiased evaluation of the model's generalization capabilities to new scenarios.

3.1.2. Object Detection Module

The trade-offs in parameter size, accuracy (mAP@50-95 on COCO), and speed on an NVIDIA T4 GPU are shown in Table 1 for popular object detection models. Faster R-CNN achieves stable accuracy but typically runs more slowly due to its two-stage architecture. SSD focuses on speed with fewer layers but sacrifices some accuracy. Meanwhile, RTMDet and RT-DETR are newer single-stage models that are good for both accuracy and real-time performance. However, a flexible range of model sizes and a good balance between speed

and accuracy are provided by the YOLO family of models, making them well recognized. This single-stage approach processes an image in one pass, making it well suited for real-time or embedded scenarios.

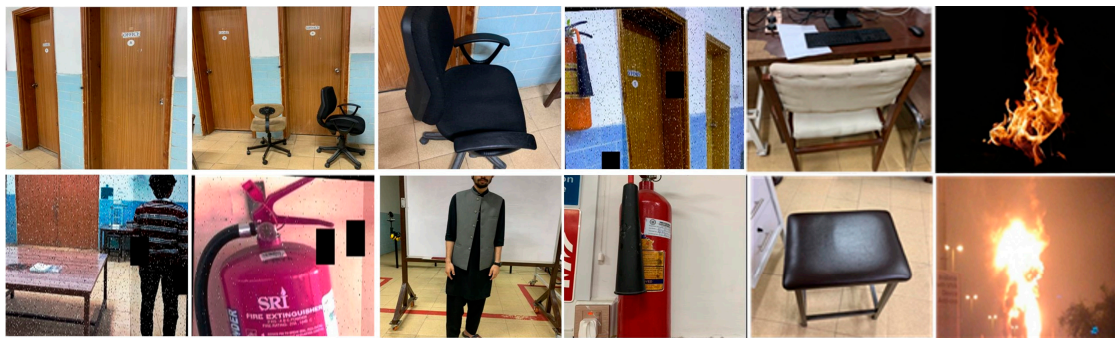


Figure 3. Indoor dataset images. Some randomly chosen images from our dataset including five interactive objects, i.e., Door, Chair, Fire Extinguisher, Person, and Fire.

Table 1. Performance comparison of object detection models on COCO Dataset.

Model	Parameters (M)	mAP@50-95 (COCO)	Speed on T4 (FPS)
RTMDet [30]	4.8–94.9	41–52.8	10–40
RT-DETR [31]	20–67	46.5–54.8	15–40
Faster R-CNN [32]	19	37.8–39	8–12
SSD [33]	22–25	26.8	45
YOLOv3-v7 [34]	1.6–151.7	37.5–57.2	30–40

For the proposed application, it is also desirable to make it cost effective; hence, Jetson Nano is chosen for hardware instead of Jetson Orin, etc., due to higher cost. Similar to other detection models, YOLOv8 and higher versions experience compatibility issues on Jetson Nano due to the higher number of parameters. Two datasets having 1000 and 3000 images respectively were trained on YOLOv7t, YOLOv5n, and SSD and assessed in terms of detection accuracy (mAP@50, mAP@50-95), inference latency, FPS, parameter count, and model size. Finally, the top-performing models were deployed on Jetson Nano to evaluate real-time feasibility in embedded environments. Based on a comprehensive trade-off between accuracy and computational efficiency, YOLOv5n was selected as the optimal model, which is discussed in Section 4.4. Among different versions of YOLOv5, YOLOv5n is a smaller and faster version which offers high inference speed and a high frame rate due to its reduced number of parameters, which are 1.9 million. This comes at the cost of lower accuracy with mAP@50 as 45.7% and mAP@50-95 as 28% on the COCO dataset. However, we enhanced the accuracy (mAP@50 as 84.5% and mAP@50-95 as 59.3%) for our dataset through hyperparameter tuning and transfer learning.

The overall schematic of the computer vision module is shown in Figure 4, which shows the several key steps required in order to detect five different kinds of interactive objects in a given scene of an indoor environment. The YOLOv5 nano (YOLOv5n) model detects objects through three steps: the backbone which extracts initial features, the neck which improves the localization and detection of objects by fusing multiscale features, and the head which calculates confidence and bounding box regression.

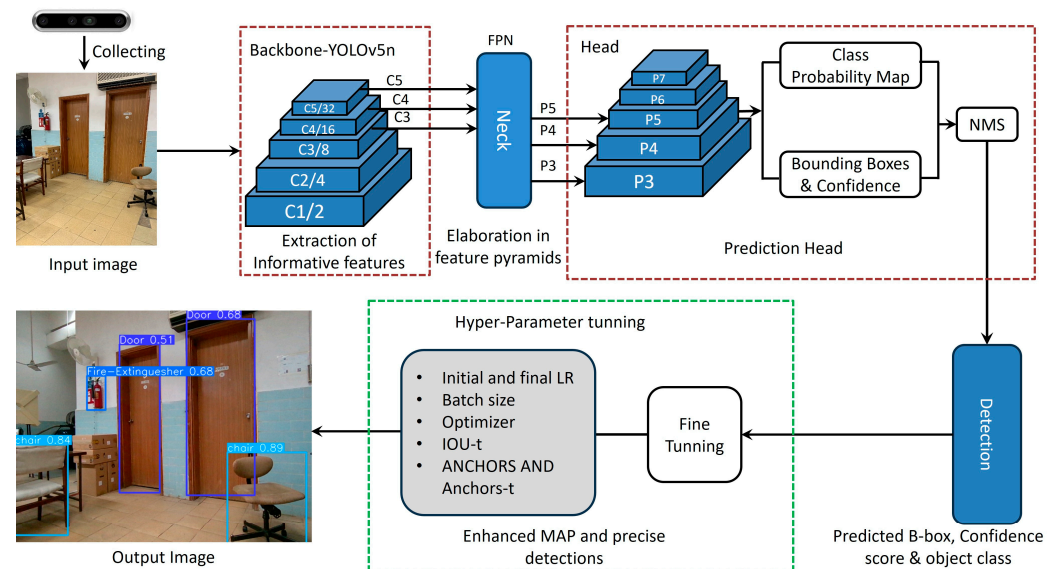


Figure 4. Overall schematic of object detection module. YOLOv5n model with hyperparameter tuning.

3.2. Distance Module

The same Intel RealSense D455 camera, used for object detection, also provides real-time depth data for each pixel in the frame, with a depth accuracy of approximately 2 mm at a distance of 1 m, where each pixel value represents the distance from the camera in millimeters. The RealSense SDK outputs this as a 16-bit depth frame. To make the depth information easier to interpret, color mapping is applied. In the default RealSense Viewer, closer objects appear in cooler colors (blue), while farther objects appear in warmer colors (red), creating a clear and intuitive visualization of depth.

The depth frame is converted to a NumPy array for efficient random access in the code which has the same height and width dimensions as the RGB image frames (640×480). The index of element in the depth array directly corresponds to an x, y pixel coordinate in the 2D image. This creates a mapping between the 2D pixel space and 3D real-world measurements. The distance module is seamlessly integrated into the overall system to ensure efficient working of the system.

Table 2 compares various distance sensing technologies including ultrasonic, infrared, laser, and radar [35] against the Active Stereo-IR Depth Camera in terms of operational principles, range, bandwidth, and accuracy, which shows the effectiveness of the Active Stereo-IR depth sensing method.

Table 2. Comparison of depth sensing technologies.

	Active Stereo-IR Depth Camera [36]	Ultrasonic [37]	Infrared [38]	Laser [39]	Radar [40]
Principle of Operation	Stereo vision with active infrared (IR) projection	Transmission and reception of acoustic waves	Transmission and reception of IR light pulses	Transmission and reception of light waves	Transmission and reception of microwaves
Range	Short range: 0.3–6 (cm); Long range: Up to 10 (m)	2–400 (cm)	20–150 (cm)	Short range: 15–120 (cm); Long range: 10 to 50 (m)	Short range: 0–200 (m); Long range: >3000 (m)
Bandwidth	Wide	Wide	Quite narrow	Narrow	Depends on antenna size
Accuracy	High	Low	HIGH	HIGH	Medium

3.3. Voice Module

The voice module is used to provide safe navigation instructions to visually impaired individuals. Once interactive objects are detected and the distance of all objects in the scene is obtained, the voice module is activated. It announces the name of various interactive objects like chair, door, fire, fire extinguisher, and person when they fall within the predefined threshold distance. The voice module also provides information about any kind of obstacle when it crosses the predefined threshold. An open-source Python3 library called Pyttsx3 is used for text-to-speech conversion and playback.

3.4. Overall System Working

The complete initial training is performed using Nvidia RTX 3060 GPU and the Windows 11 platform. Next, the trained model is embedded into the NVIDIA Jetson Nano to perform wearable mobility assistance to visually impaired individuals. The proposed experimental setup involves integrating several hardware components which include an NVIDIA Jetson Nano, an Intel RealSense D455 depth camera, a power bank, and a JBL mini-Bluetooth speaker. The NVIDIA Jetson Nano serves as the central processing unit due to its high computational power and onboard GPU capabilities. The Intel RealSense D455 depth camera is connected to the Jetson Nano via USB 3.0 which captures both RGB and depth images. Power to all components is provided by a power bank with a capacity of 20 watts with 5 volts and 4 amperes of current which is connected to the Jetson Nano through the Barrel Power Jack. The overall system draws approximately 12–18 W during active sensing and processing, which implies an operational duration of roughly 5–8 h on a 20,000 mAh power bank. For audio feedback, a wire JBL mini-Bluetooth speaker is connected to the Jetson Nano via an external Bluetooth adapter. The detailed hardware integration with NVIDIA Jetson Nano is shown in Figure 5a. The overall wearable system worn by the visually impaired individual is shown in Figure 5b, which shows the portability of the system. This setup ensures that the user receives real-time voice alerts about detected objects, obstacles, and their distances. The entire hardware setup is mounted onto a wearable belt which makes it comfortable and easy for the user to wear and walk naturally. The complete integration and working of the proposed system are shown in Figure 6. The depth camera continuously captures real-time images and depth information of the user's surroundings. These data are then passed on to the Jetson Nano, which processes it through several stages. The first stage involves the trained YOLOv5n model, which identifies and classifies all the interactive objects within the images. Once the objects are detected, the system integrates the depth information with the detection results. This step is crucial as it allows the system to not only detect objects but also understand their spatial distances from the user. The output generated by the system includes two main components: an object detection image where detected objects are highlighted, and a depth image that represents the distances of these objects from the camera which is installed on the user's chest. If interactive objects are detected, then the system evaluates the distance of the detected objects. If this distance crosses the predefined threshold like 1.4 m to 1.6 m, then audio feedback is provided to the user about the detected objects and their distances via Bluetooth speaker, e.g., a chair is detected at 1.5 m, etc. Moreover, if no interactive object is detected but the distance of obstacles is detected within the predefined threshold based on the depth image, then audio feedback of obstacle detection at 1.5 m is provided to the user via Bluetooth speaker. The whole system helps the user to navigate in indoor environment safely. This real-time feedback loop is essential for the user, as it continuously updates about potential obstacles and surroundings, enhancing their spatial awareness and mobility. If the threshold distance is set too high, more nearby objects will be announced, which will result in overloaded auditory feedback and cause inconvenience to users in assistive

navigation. On the other hand, if the threshold is too low, visually impaired users might not have enough time to stop or change their path. Therefore, the threshold is adjustable, allowing users to customize it based on their individual needs.

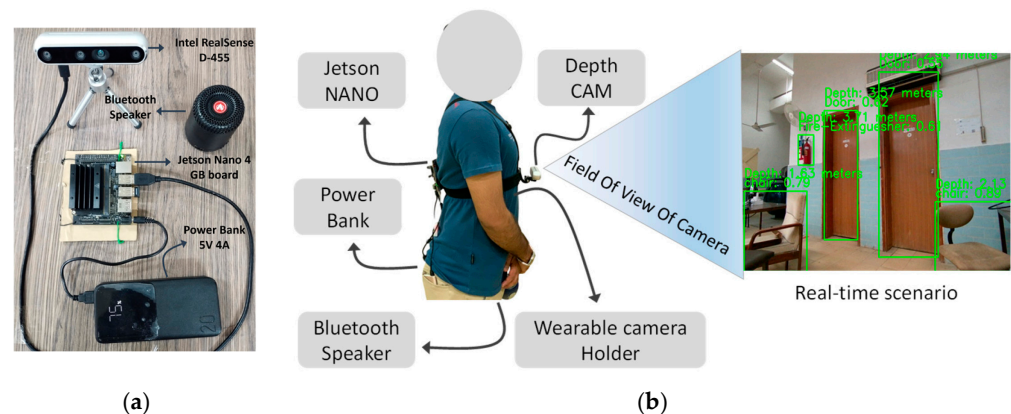


Figure 5. Experimental setup. The overall experimental setup with (a) the hardware integration with Nvidia Jetson Nano and (b) the overall setup worn by the visually impaired individual for navigating the indoor environment.

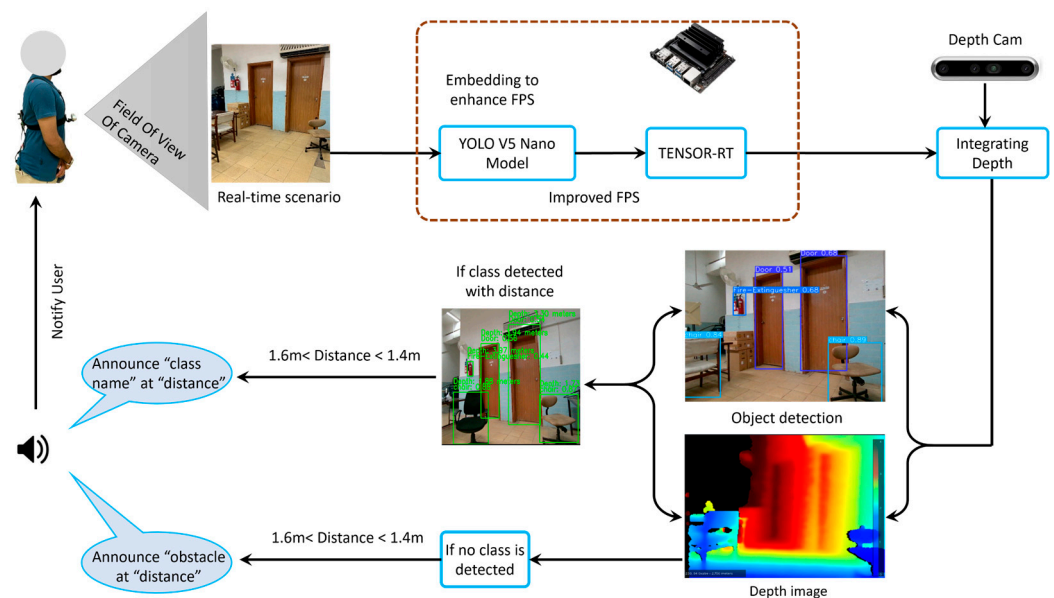


Figure 6. Detailed flow diagram of integrated system with all modules. The computer vision module detects all interactive objects, the distance module finds the distance of the user from all the interactive objects and the obstacles, while the voice module give an auditory feedback to the user about the interactive objects and obstacle and their distances from the user.

4. Performance Evaluation of the Proposed System

The performance of the overall system mainly depends on two modules: the vision module and the distance module, so the results and performance evaluation of these modules are discussed in the following sub-sections.

4.1. Optimization

4.1.1. Hyperparameter Tuning

The pre-trained YOLOv5n on the MS-COCO dataset did not show satisfactory results for indoor assistive navigation applications. The model provides enough FPS; however, object detection performance is below par. Therefore, the model is fine-tuned multiple

times using different combinations of hyperparameters to achieve the desired results. Hyperparameter tuning involves careful adjustment of various hyperparameters like learning rates, batch sizes, anchors, etc., which effectively helps to improve the model's performance on specific tasks, such as object detection. The number of classes is increased gradually to improve the performance of the YOLOV5n model. Initial hyperparameter tuning was focused on object detection for two classes. By fine-tuning various hyperparameters, an mAP@50 of 0.98 was achieved. The parameter combinations such as larger batch sizes with cosine learning rate decay also contributed to achieving this mAP. The SGD optimizer outperformed Adam in minimizing the loss function, as SGD performs a parameter update for each image or mini batch, while Adam computes adaptive learning rates for each parameter. Higher anchor thresholds required tighter correspondence between predictions and ground truth boxes, reducing incorrect matches and improving learning. Similarly, the hyperparameters were tuned for three class problems to find the most suitable hyperparameter combination to be used as a starting combination for five class problems. The complete hyperparameter tuning process for five classes of detection tasks is shown in Table 3. After hyperparameter tuning, an mAP@50 of 0.885 on the validation dataset is achieved. Initially, the original YOLOV5n has only a 0.457 mAP@50 for the MS-COCO dataset.

Table 3. Optimization of hyperparameters for five classes.

S. No	Initial-Lr	Final-Lr	Anchors_t	Batch Size	Optimizer	Lr Scheduler	mAP@50
1	0.001	0.00001	3.95	8	SGD	Cos_lr	0.70
2	0.001	0.00001	3.95	16	Adam	Cos_lr	0.65
3	0.001	0.00001	3.9	8	SGD	Cos_lr	0.77
4	0.001	0.00001	3.9	8	Adams	Cos_lr	0.73
5	0.001	0.00001	4	8	SGD	Cos_lr	0.841
6	0.001	0.00001	4	8	SGD	LRRD	0.885

4.1.2. Optimization of Overall System from User's Perspective

The overall system is also optimized from the user's perspective, such as by determining the optimal threshold distance and clear auditory feedback text, etc. The no threshold condition results in announcing all the objects detected within the camera's field of view. This approach produced an excessive number of voice announcements, which users found distracting. In a subsequent trial, a larger threshold was implemented, with announcements made for all detected class names and obstacles within that range. However, user feedback indicated that the high volume of announcements is still confusing. Moreover, during the experiment with a short threshold distance, the users reported that the brief announcement range did not allow sufficient time to respond, interact with assistive objects, or avoid obstacles, etc. Accordingly, the distance between 1.4 to 1.6 m was found to be the optimal threshold, considering normal human walking. In our experiment, a 1.5 m threshold is used; moreover, this threshold is kept adjustable in the proposed system to accommodate the user's personal preference. Additionally, too long verbal phrases in announcements also cause inconvenience to users. For instance, announcing "Chair is detected at 1.5 m" was found to be inconvenient. So, the announcement length was reduced, such as "Chair at 1.5 m" or simply announcing the name of the class like "chair", "person", and "obstacle", etc. was found to be more appropriate as the user already knows that the system is announcing the name of an object which is within the threshold distance. In the results Section 5.1, multiple types of auditory announcements are made to show the effectiveness of the proposed model. However, for safety purposes, the class "fire" is announced regardless of the threshold distance.

4.2. Object Detection: Training and Performance Evaluation

In this section, the training and performance of the object detection model is discussed in detail. Figure 7a,b show the training loss over epochs for two metrics: box loss (accuracy of bounding box predictions) and class loss (accuracy of object classification) for both the training and validation data. Both losses decreased rapidly during the first 50 epochs as the model quickly learns data patterns. Between 50 and 181 epochs, the decline slows, and eventually, the model nearly converges with box loss around 0.02 and class loss around 0.003, indicating high confidence in its predictions.

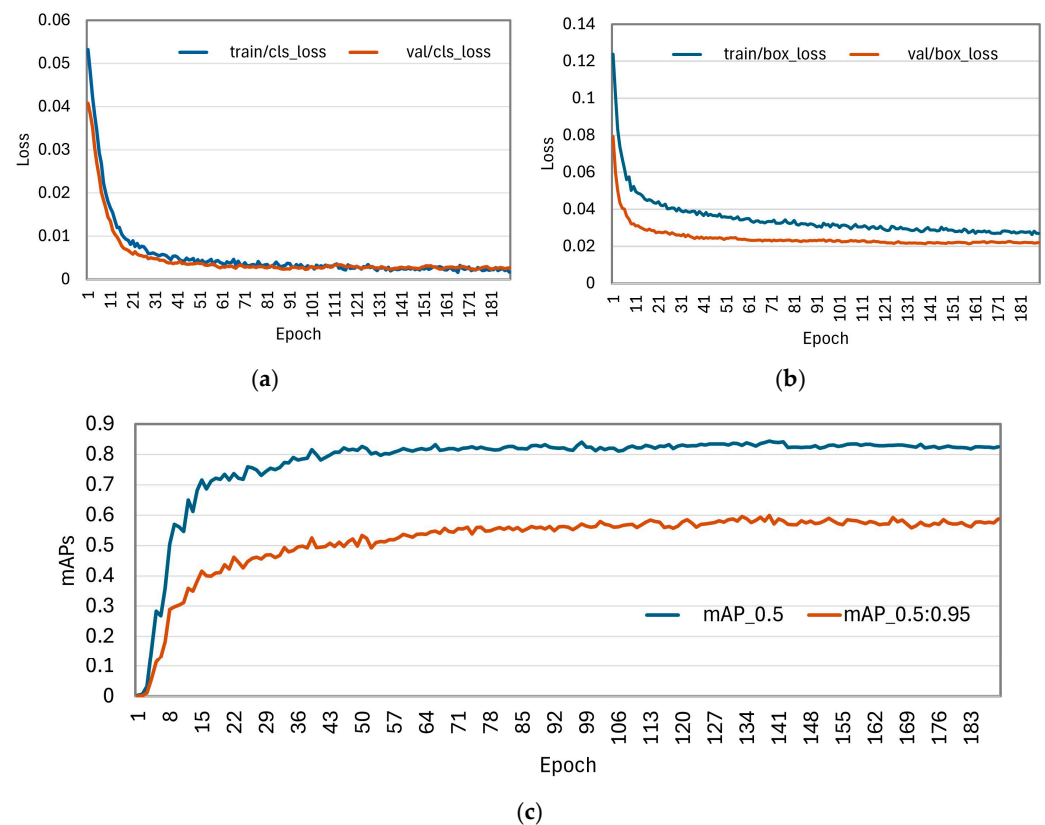


Figure 7. (a) Graph of class loss vs. epoch for training and validation data. (b) Graph of box loss vs. epoch for training and validation data. (c) Graph of mAP50 and mAP50-95 vs. epoch for bounding box.

The mAP@50 and mAP@50-95 measure the mean average precisions at a 50 IoU threshold, while the mAP@50-95 measures the mean average precisions over thresholds ranging from 50 to 95 (in 0.05 increments). As shown in Figure 7c, the mAP@50 is higher due to its less stringent criteria during processing. The model learns rapidly in the first 50 epochs, slows from 50 epochs, and stabilizes with the mAP@50 settling at around 0.845 and the mAP@50-95 around 0.593.

To check the performance, the model is evaluated on the unseen test dataset. The trained model achieves a 0.845 value of mAP@50 which shows the model effectiveness in detecting five different types of interactive objects. The average precision metric (mAP) for all five interactive objects is shown in Table 4. The table provides precision, recall, and both AP@50 and AP@50-95 for all five types of interactive objects. The high precision, recall, and average precision values for all classes indicate that the model's performance is considered good in detecting different interactive objects. Lower values of mAP at a high IoU threshold are expected for our model.

Table 4. Precision, recall, mAP@50, and mAP@50-95 for all classes.

Names	Classes	Precision (P)	Recall (R)	mAP@50	mAP@50-95
Door	0	0.95	0.597	0.82	0.535
Fire	1	0.953	0.762	0.881	0.591
Fire	2	0.898	0.977	0.929	0.719
Extinguisher					
Chair	3	0.75	0.707	0.754	0.475
Person	4	0.821	0.913	0.84	0.644

4.3. Real-Time FPS and Distance Module Performance Evaluation

The real-time FPS drops after embedding the integrated model on the Jetson Nano. The drops in FPS occur due to the integration of the three modules which increased the overall computational cost of the system. Primarily, we identified that the delay was caused by the object detection algorithm in real time due to the limited processing power of the Jetson Nano. To overcome this issue, the weight of the trained object detection algorithm is embedded using TENSOR-RT optimization which significantly improved the FPS from 5 to 19 FPS on the Jetson Nano in real time. Moreover, the evaluation of the distance estimation module showed neglectable deviation between calculated distances and actual distances, which is almost 0% within the 0.5 m and 2.16% at the 10 m range. The detailed tables which show the FPS and distance module performance are provided in the Supplementary Materials.

4.4. Comparative Evaluation of Object Detection Models

To presents a comprehensive comparative analysis of SSD, YOLOv5n, and YOLOv7tiny across different dataset scales and hardware platforms, the evaluation focuses on detection accuracy, localization precision, computational efficiency, and real-time deployment feasibility. Experiments were conducted on NVIDIA RTX 3060 GPU and Jetson Nano embedded hardware using standardized metrics, including mAP@50, mAP@50-95, inference latency, FPS, parameter count, and model size.

The initial experiments were conducted on a dataset comprising 1000 images to investigate the baseline performance of the selected models. As reported in Table 5, YOLOv5n achieved an mAP@50 of 0.845 and an mAP@50-95 of 0.593, outperforming both YOLOv7tiny and SSD. In comparison, YOLOv7tiny obtained mAP@0.5 and mAP@50-95 values of 0.584 and 0.325, respectively, while SSD achieved 0.6219 and 0.4792. Consequently, YOLOv5n exhibited a relative improvement of 44.7% in mAP@50 over YOLOv7tiny and 35.9% over SSD. Similarly, in terms of mAP@50-95, YOLOv5n improved performance by 82.5% over YOLOv7tiny and 23.7% over SSD. From a model complexity perspective, YOLOv5n required only 1.17 million parameters and occupied 3.38 MB of storage, whereas YOLOv7tiny and SSD required 6.02 million parameters (12 MB) and 24.3 million parameters (94.35 MB), respectively. This corresponds to an 80.6% reduction in parameter count relative to SSD and an 80.5% reduction relative to YOLOv7tiny, highlighting the architectural efficiency of YOLOv5n. In terms of inference efficiency, YOLOv5n achieved an inference latency of 2.9 ms and an FPS of 75, compared to 3.8 ms and 60 FPS for YOLOv7tiny, and 10.93 ms and 22 FPS for SSD. Thus, YOLOv5n reduced inference latency by 73.5% and increased FPS by 241% compared to SSD, demonstrating its suitability for real-time applications. However, given the limited dataset size, the representational capacity of larger models such as SSD and YOLOv7tiny may not have been fully exploited. Therefore, the dataset was expanded to ensure a more reliable and statistically meaningful comparison.

Table 5. Comparison among YOLOv5, YOLOv7t, and SSD on GPU.

Model	@1000 Images Dataset			@3000 Images Dataset		
	YOLOv5n	YOLOv7tiny	SSD	YOLOv5n	YOLOv7tiny	SSD
mAP@50	0.845	0.584	0.6219	0.907	0.796	0.6877
mAP@50-95	0.593	0.325	0.4792	0.674	0.5	0.4921
Inference time (ms)	2.9	3.8	10.93	2.9	3.8	10.93
FPS	50	43	60	50	43	60
Parameters (M)	1.176	6.025	24.3	1.176	6.025	24.3
Model size (MB)	3.38	12.0	94.35	3.38	12.0	94.35

To evaluate the scalability and generalization capability of the models, the dataset size was increased to 3000 images, and all models were retrained under identical experimental conditions. The results, summarized in Table 5, confirm the robustness of YOLOv5n. On the enhanced dataset, YOLOv5n achieved an mAP@50 of 0.907 and an mAP@50-95 of 0.674. In contrast, YOLOv7tiny achieved 0.796 and 0.500, while SSD achieved 0.6877 and 0.4921. Quantitatively, YOLOv5n outperformed YOLOv7tiny by 13.9% and SSD by 31.9% in mAP@50. Similarly, YOLOv5n achieved improvements of 34.8% and 37.0% in mAP@50-95 over YOLOv7tiny and SSD, respectively. These results indicate that YOLOv5n not only maintains superior accuracy but also exhibits stronger data scalability. Notably, inference latency and FPS remained constant across dataset sizes, implying that YOLOv5n preserves its computational efficiency irrespective of dataset scale, which is critical for real-time systems.

To further analyze the detection characteristics of the models, a class-wise evaluation was performed on the 3000-image dataset (Table 6). The results demonstrate that YOLOv5n consistently achieves higher accuracy across most object categories. For example, YOLOv5n achieved mAP@50 values of 0.886 (Door), 0.917 (Fire), 0.931 (Fire Extinguisher), and 0.958 (Chair). Compared to SSD, YOLOv5n improved detection accuracy by 144.8% (Door), 53.6% (Fire), 44.1% (Chair), while showing comparable performance for Fire Extinguisher. Moreover, the mAP@50-95 metric reveals that YOLOv5n provides superior localization precision, particularly for objects with complex geometries and scale variations. This suggests that YOLOv5n benefits from more effective multi-scale feature fusion and bounding-box regression mechanisms. The class-wise analysis further indicates that SSD exhibits relatively lower robustness across classes, while YOLOv7tiny shows higher inter-class performance variance, reflecting limitations in feature representation and localization accuracy.

Table 6. Class-wise mAPs of SSD model @ 3000-image dataset on GPU.

S. No	Classes	SSD		Yolov5n		Yolo7t	
		mAP@50	mAP@50-95	mAP@50	mAP@50-95	mAP@50	mAP@50-95
1	Door	0.362	0.2673	0.886	0.701	0.804	0.597
2	Fire	0.5969	0.4580	0.917	0.524	0.903	0.408
3	Fire Extinguisher	0.9434	0.6970	0.931	0.724	0.635	0.476
4	Chair	0.6650	0.4238	0.958	0.787	0.893	0.626
5	person	0.8710	0.6144	0.842	0.635	0.743	0.412

To assess the feasibility of real-time deployment in resource-constrained environments, YOLOv5n and YOLOv7tiny were evaluated on the Jetson Nano platform (Table 7). These models were selected based on their superior performance on GPU. YOLOv5n achieved an inference time of 52.6 ms and an FPS of 19, which decreased to 12 FPS when system-level

delays were considered. In contrast, YOLOv7tiny achieved an inference time of 76.9 ms and an FPS of 13, dropping to 5 FPS with delays. Quantitatively, YOLOv5n reduced inference latency by 31.6% and improved FPS by 46.2% compared to YOLOv7tiny. When system-level delays were included, YOLOv5n achieved a 140% higher effective FPS, highlighting its superior suitability for embedded real-time applications. These findings confirm that model compactness and architectural efficiency play a critical role in achieving real-time performance on low-power hardware platforms.

Table 7. Real-time performance comparison of YOLOv5 and YOLOv7t on Jetson Nano.

S. No	Model	Inference Time (ms)	FPS (Model Inference Only)	FPS (Including Delays)
1	YOLOv5n	52.6	19	12
2	YOLOv7t	76.9	13	5

The experimental results consistently demonstrate that YOLOv5n achieves an optimal balance between detection accuracy, localization precision, and computational efficiency. Its superior performance can be attributed to its lightweight backbone, efficient feature pyramid network, and optimized anchor-based detection head, which collectively enhance feature representation while minimizing computational overhead. In contrast, SSD suffers from limited multi-scale feature extraction capability, leading to reduced localization accuracy, while YOLOv7tiny, despite its improved architectural design, incurs higher computational complexity without proportional accuracy gains. Therefore, the results indicate that YOLOv5n is the most suitable model for real-time object detection in both high-performance GPU environments and embedded systems. Its consistent superiority across datasets, metrics, and hardware platforms underscores its robustness and practical applicability in real-world vision-based applications.

5. Real-Time Implementation and Comparative Analysis

5.1. Real-Time Implementation of Proposed Assistive Navigation System

To validate the complete performance of the proposed system, real-time experimentation is performed in an indoor environment. Different interactive objects and obstacles are placed at random positions in an indoor environment to check the performance of the proposed system. The participants voluntarily took part in the real-time experiment and provided their consent for participation.

Some snips from video results are shown in Figure 8. The results show a detailed description of both external and internal camera results at different time frames. The top images display varied scene frames captured over time by the external camera system. Corresponding bottom images show the output of the depth cam detecting objects in real time, respectively. At the 5 s time frame, an obstacle is detected as the user entered the distance threshold range, which triggered the audio prompt “Obstacle”. The system provides auditory feedback to the user through the speaker. Instead of announcing “Obstacle is detected at 1.5 m” the system just announced “Obstacle” to provide brief information about the environment to the user. Subsequently at 9 s, a chair is detected by the vision module and after reaching the threshold, the system provides the audio feedback “Chair” to the user. At 12 s, another chair entered the field of view and fell within the threshold, and the system provided auditory feedback to the user. At 15 s, an obstacle is detected within the threshold distance, and accordingly the audio feedback “Obstacle” is provided to the user. As seen at the 18 s frame, two doors and a person were identified concurrently by the object detection module. However, only the closer object “Person” is notified to the

user, based on threshold parameter. A similar case occurred at 25 s, where door detection crossed the threshold distance, and the user was notified.



Figure 8. Real-time experimentation results of the smart assistive navigation system at different time frames.

The real-time testing by different users and in different environments was performed to check the generalization of the model. The results illustrate both the external and internal camera outputs at different time intervals. At the 104, 108, 114, 117, 125, 132, 136 s time frames, an obstacle, door, chair, person, chair, obstacle, and door are detected, respectively, as shown in Figure 9. At the 204, 207, 211, 216, 227 s time frames, a chair, person, chair, obstacle, and door are detected, respectively, as shown in Figure 10.

We observe a small delay in the real-time streaming video which is due to the transmission of the video to remotely connected systems. The streaming is done to record the video to showcase the internal cam view results as shown in the attached videos. However, real-time video streaming is not required to assist a visually impaired individual. By skipping the streaming step, delays can be avoided which allow the Jetson Nano to process the algorithm more quickly, thus reducing the latency further for real-time applications.

In summary, real-time experimental results demonstrate the promising performance of the proposed integrated framework. This validates the real-time implementation of the proposed systems and the potential to extend this work towards developing additional assistive technologies to support outdoor navigation scenarios and aid those with diverse disabilities. The successful implementation of the proposed system will certainly benefit the daily life activities of the visually impaired individual.



Figure 9. Real-time experimentation results of the smart assistive navigation system at different time frames for different users and environments.

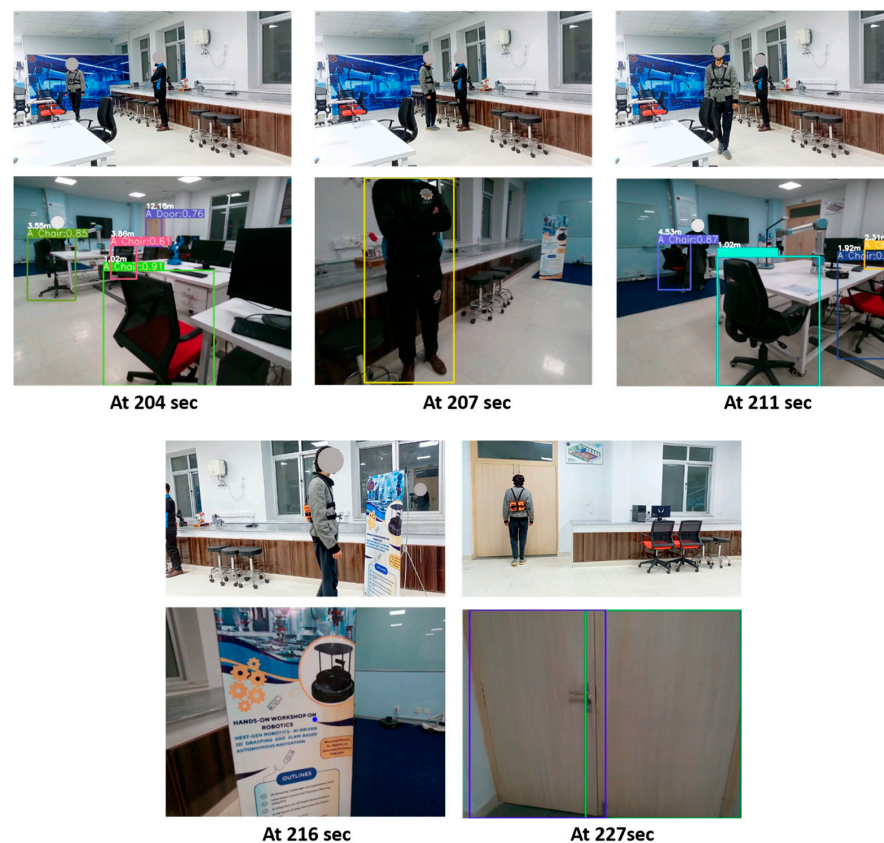


Figure 10. Real-time experimentation results of the smart assistive navigation system at different time frames for different users and environments.

5.2. Quantitative Analysis in Real-Time Navigation

Table 8 presents the step-wise as well as overall latency (in milliseconds) for different object-based announcements generated by the system. For each announcement type, the processing pipeline consists of frame capture, YOLO inference, post-processing, drawing, and the final announcement time. The first four steps—frame capture (9.13 ms), YOLO inference (52.6 ms), post-processing (0.9 ms), and drawing (2.7 ms)—remain constant across all cases regardless of the detected object. The variation in total time is primarily due to the announcement time, which depends on the length of the spoken message. For example, shorter announcements such as “A door” result in a lower total latency (965.33 ms), while longer messages like “A fire extinguisher” or “Obstacle detected” lead to higher total times of 2065.33 ms and 1765.33 ms, respectively. Overall, the table highlights that the dominant contributor to system latency is the announcement generation, whereas the visual detection and processing stages contribute only a small and consistent delay.

Table 8. Step-wise and overall latency in milliseconds.

Sr. #	Announcement Type	Frame Capture	Yolo Inference	Post Processing	Drawing	Announcement Time	Total Time
1	A person	9.13	52.6	0.9	2.7	1100	1165.33
2	A door	9.13	52.6	0.9	2.7	900	965.33
3	A chair	9.13	52.6	0.9	2.7	1000	1065.33
4	A fire	9.13	52.6	0.9	2.7	1100	1165.33
5	A fire extinguisher	9.13	52.6	0.9	2.7	2000	2065.33
6	Obstacle detected	9.13	52.6	0.9	2.7	1700	1765.33

Table 9 summarizes the performance of an object detection and announcement system across seven experiments. For each experiment, it reports how many objects appeared, how many were missed at a given instant, the resulting object detection accuracy, and how successfully detected objects were announced, measured by the Announcement Completion Rate (ACR). Across all experiments, no objects were completely missed overall, and detection accuracy ranged roughly from 86% to 93%. The system achieved an overall object detection performance of 88.64% and a high overall ACR of 93.06%, indicating reliable detection and announcement performance with only minor variations between experiments.

Table 9. Real-time navigation performance evaluation.

Exp. No.	Overall Missing Objects	Objects Missed at an Instant	Total Objects Appeared	Object Detection (%)	Successful Announcement	Total Objects	Announcement Completion Rate (ACR) (%)
1	0	2	22	90.91	6	7	85.71
2	0	3	23	86.96	7	7	100.00
3	0	2	15	86.67	5	5	100.00
4	0	2	18	88.89	6	7	85.71
5	0	2	16	87.50	8	8	100.00
6	0	2	15	86.67	4	5	80.00
7	0	1	14	92.86	5	5	100.00
Overall Performance				88.64%	--	--	93.06%

5.3. Comparison with Existing Solutions in Terms of Technological Advancement

In order to evaluate the capabilities of the proposed system, a qualitative comparative analysis with existing devices is provided in this section. For comparison, different capabilities, including obstacles and interactive object detection, depth estimation, audi-

tory feedback, wearability, and real-time testing/implementation, are considered. Mostly, the existing assistive devices cover basic obstacle detection, as seen with tools like Smart Cans and GPS Cane, etc. There are few devices available in the literature which have utilized the objective detection model along with an auditory feedback system. Table 10 shows the comparison with existing state-of-the-art devices. The integrated features, especially the detection of obstacles and interactive objects with depth estimation, address current gaps, enhance situational awareness, and support independent navigation for visually impaired users.

Table 10. Comparative analysis of assistive devices.

Assistive Tool	Obstacle Detection	Interactive Objects Recognition	Depth Estimation	Verbal Feedback	Wearable	Real-Time Testing
Smart Cans [35,41,42]	Yes	No	Yes	No	No	Yes
GPS Cane [43]	Yes	No	Yes	No	No	No
ES (Jetson Nano) [44]	No	Yes	No	Yes	Yes	No
Proposed System	Yes	Yes	Yes	Yes	Yes	Yes

5.4. Future Work and Scalability of Proposed System

The promising performance of the proposed system opens exciting opportunities for further enhancement. The proposed system is validated in an indoor environment by considering the five classes as interactive objects, which include the Door, Chair, Fire Extinguisher, Person, and Fire. In addition, the proposed system is capable of detecting any kind of object when it falls under a predefined threshold distance. In future work, the scope of the work would be extended by increasing the number of classes for interactive objects. The model would be trained using both indoor and outdoor datasets including stairs, footpath, car, and bike, etc. Moreover, the reasoning capability of the proposed systems would be enhanced based on user requirements. For instance, after detecting the table, the system asked the user if he/she wanted to scan the tabletop. The upgraded systems would be able to detect a water bottle, keys, remote control, and other interactive objects on the tabletop. Additionally, assistance in navigation by integrating GPS, especially in outdoor environments, would also enhance the independence of visually impaired people. Moreover, potential enhancements in currency recognition and currency counting are also being considered to expand the applications of the proposed system to assist visually impaired people.

6. Conclusions

In this study, a voice-enabled along with auditory feedback system is proposed to assist visually impaired people in indoor environments. The system consists of three modules: vision, distance calculation, and voice modules which are embedded into Jetson Nano for real-time testing. The vision module consists of the YOLOv5n model which is trained to detect an object with high accuracy. Despite the reduction in parameters in YOLOv5n compared to other models, which typically affects accuracy, the fine-tuning of hyperparameters significantly enhanced the model's performance, achieving an mAP@50 of 88.5%. Additionally, the initial frame rate of 5 FPS was increased to 19 FPS after deploying all three modules with Tensor-RT optimization for real-time testing. The accuracy of the distance module is excellent with nearly 0% deviation from actual measurements within the 1.5 m threshold, which is very critical for indoor assistance. The voice module successfully conveys the auditory feedback to the user about the presence of interactive objects and

obstacles and their distances. The developed system can effectively be used for full-day activities to assist the visually impaired individual in daily life with greater independence.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/automation7020041/s1>. Demonstrational experimental video, along with tables for distance module performance evaluation and speed, are provided as Supplementary Materials.

Author Contributions: Conceptualization, A.I., K.K., A.A. and S.M.; methodology, S.S.S., S.-U.-R. and M.A.; software, S.S.S., S.-U.-R. and M.A.; validation, A.I., K.K. and G.J.S.; analysis, A.I., A.A. and K.K.; investigation, S.S.S., G.J.S. and M.A.; data curation, S.-U.-R. and M.A.; experimentation, S.-U.-R. and S.S.S.; writing—original draft preparation, S.S.S. and S.-U.-R.; writing—review and editing, A.I., K.K., A.A. and S.M.; visualization, S.S.S., A.I. and A.A.; resources, A.I., S.M. and G.J.S.; supervision, A.I. and S.M.; project administration, A.I.; funding acquisition, A.I. and G.J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Higher Education Commission (HEC) Pakistan under NRPUPROJECT 20-14813.

Data Availability Statement: Data would be provided upon request.

Acknowledgments: Authors thank the GIK Institute and Higher Education Commission (HEC) Pakistan for providing support to this research under NRPUPROJECT 20-14813.

Conflicts of Interest: Authors declare that they have no competing interests.

Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You only look once
SGD	Stochastic gradient descent
mAP	Mean average precision
FPS	Frames per second

References

1. Ramchandran, R.S.; Hindman, H.B.; Sørensen, S. Vision Impairment and Its Management in Older Adults. In *Geriatric Rehabilitation*; CRC Press: Boca Raton, FL, USA, 2017; pp. 293–307.
2. Munemo, E.; Tom, T. Problems of unemployment faced by visually impaired people. *Greener J. Soc. Sci.* **2013**, *3*, 203–219. [\[CrossRef\]](#)
3. Jeamwatthanachai, W.; Wald, M.; Wills, G. Indoor navigation by blind people: Behaviors and challenges in unfamiliar spaces and buildings. *Br. J. Vis. Impair.* **2019**, *37*, 140–153. [\[CrossRef\]](#)
4. Rosenberg, E.A.; Sperazza, L.C. The visually impaired patient. *Am. Fam. Physician* **2008**, *77*, 1431–1436. [\[PubMed\]](#)
5. Ardiansah, J.T.; Okazaki, Y. An interactive self-learning system using smartphone app and cards enabling braille touch experience for blindness. *Br. J. Vis. Impair.* **2024**, *43*, 694–706. [\[CrossRef\]](#)
6. Theodorou, P.; Meliones, A.; Filios, C. Smart traffic lights for people with visual impairments A literature overview and a proposed implementation. *Br. J. Vis. Impair.* **2023**, *41*, 697–725. [\[CrossRef\]](#)
7. Garaj, V.; Jirawimut, R.; Ptasinski, P.; Cecelja, F.; Balachandran, W. A system for remote sighted guidance of visually impaired pedestrians. *Br. J. Vis. Impair.* **2003**, *21*, 55–63. [\[CrossRef\]](#)
8. Liao, C.-F. *An Integrated Assistive System to Support Wayfinding and Situation Awareness for People with Vision Impairment*; University of Minnesota: Minneapolis, Minnesota, 2016.
9. Tahoun, N.; Awad, A.; Bonny, T. Smart assistant for blind and visually impaired people. In *Proceedings of the 3rd International Conference on Advances in Artificial Intelligence, Istanbul, Turkey, 26–28 October 2019*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 227–231.
10. Kim, I.-J. Recent advancements in indoor electronic travel aids for the blind or visually impaired: A comprehensive review of technologies and implementations. *Univers. Access Inf. Soc.* **2025**, *24*, 173–193. [\[CrossRef\]](#)
11. Al-Smadi, A.; Al-Qaryouti, T.; Rehan, A.; Assi, H.; Alshareea, A. A navigation tool for visually impaired and blind people. *Proc. Eurasia Proc. Sci. Technol. Eng. Math.* **2023**, *22*, 119–126. [\[CrossRef\]](#)

12. Masal, K.M.; Bhatlawande, S.; Shingade, S.D. Development of a visual to audio and tactile substitution system for mobility and orientation of visually impaired people: A review. *Multimed. Tools Appl.* **2023**, *83*, 20387–20427. [[CrossRef](#)]
13. Faseeh, M.; Bibi, M.; Khan, M.A.; Kim, D.-H. Deep learning assisted real-time object recognition and depth estimation for enhancing emergency response in adaptive environment. *Results Eng.* **2024**, *24*, 103482. [[CrossRef](#)]
14. Treuillet, S.; Royer, E. Outdoor/indoor vision-based localization for blind pedestrian navigation assistance. *Int. J. Image Graph.* **2010**, *10*, 481–496. [[CrossRef](#)]
15. Khenkar, S.; Alsulaiman, H.; Ismail, S.; Fairaq, A.; Jarraya, S.K.; Ben-Abdallah, H. ENVISION: Assisted navigation of visually impaired smartphone users. *Procedia Comput. Sci.* **2016**, *100*, 128–135. [[CrossRef](#)]
16. Thakur, S.; Joshi, A.; Grover, A. Mobile Phones as Assistive Technologies: Gaps and Opportunities. In *Advances in Assistive Technologies*; Springer: Heidelberg, Germany, 2025; pp. 33–47.
17. Dümbgen, F.; Hoffet, A.; Kolundžija, M.; Scholefield, A.; Vetterli, M. Blind as a bat: Audible echolocation on small robots. *IEEE Robot. Autom. Lett.* **2022**, *8*, 1271–1278. [[CrossRef](#)]
18. Zhang, Y.; Zhang, B.; Shen, C.; Liu, H.; Huang, J.; Tian, K.; Tang, Z. Review of the field environmental sensing methods based on multi-sensor information fusion technology. *Int. J. Agric. Biol. Eng.* **2024**, *17*, 1–13. [[CrossRef](#)]
19. Okolo, G.I.; Althobaiti, T.; Ramzan, N. Assistive systems for visually impaired persons: Challenges and opportunities for navigation assistance. *Sensors* **2024**, *24*, 3572. [[CrossRef](#)]
20. Chen, X.; Zhang, Z. Optimization Research of Bird Detection Algorithm Based on YOLO in Deep Learning Environment. *Int. J. Image Graph.* **2025**, *25*, 2550059. [[CrossRef](#)]
21. Zhu, Y.; Zhang, Y.; Sun, K. A Lightweight Network for Vehicle and Pedestrian Object Detection in Autonomous Driving Scenarios. *Int. J. Image Graph.* **2025**, 185–194. [[CrossRef](#)]
22. Dubey, P.; Miller, S.; Günay, E.E.; Jackman, J.; Kremer, G.E.; Kremer, P.A. You Only Look Once v5 and Multi-Template Matching for Small-Crack Defect Detection on Metal Surfaces. *Automation* **2025**, *6*, 16. [[CrossRef](#)]
23. Gu, H.; Yoon, H.J.; Jafarnejadsani, H. Robust Object Detection Under Adversarial Patch Attacks in Vision-Based Navigation. *Automation* **2025**, *6*, 44. [[CrossRef](#)]
24. Abidi, M.H.; Siddiquee, A.N.; Alkhalefah, H.; Srivastava, V. A comprehensive review of navigation systems for visually impaired individuals. *Heliyon* **2024**, *10*, e31825. [[CrossRef](#)]
25. Chen, K.; Xiao, J.; Liu, J.; Tong, Q.; Zhang, H.; Liu, R.; Zhang, J.; Ajoudani, A.; Chen, S. Semantic visual simultaneous localization and mapping: A survey. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 7426–7449. [[CrossRef](#)]
26. Mai, C.; Chen, H.; Zeng, L.; Li, Z.; Liu, G.; Qiao, Z.; Qu, Y.; Li, L.; Li, L. A smart cane based on 2d lidar and rgb-d camera sensor-realizing navigation and obstacle recognition. *Sensors* **2024**, *24*, 870. [[CrossRef](#)] [[PubMed](#)]
27. Vennila, P.; Mangayarkarasi, V.A.; Vinayakan, K.; Raja, G.R.G. Smart IoT Navigation System for Visually Impaired Individuals: Improving Safety and Independence with Advanced Obstacle Detection. *Int. J. Comput. Res. Dev.* **2025**, *10*, 17–23.
28. Sargsyan, E. *Multimodal Interactive Interfaces for the Comprehension and Navigation of Complex Environnements by People with Visual Impairments*; Université de Toulouse: Toulouse, France, 2024.
29. Matei, M.; Alboaie, L. Enhancing Accessible Navigation: A Fusion of Speech, Gesture, and Sonification for the Visually Impaired. *Procedia Comput. Sci.* **2024**, *246*, 2558–2567. [[CrossRef](#)]
30. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. RtmDET: An empirical study of designing real-time object detectors. *arXiv* **2022**, arXiv:2212.07784. [[CrossRef](#)]
31. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs Beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024*; IEEE: New York, NY, USA, 2024; pp. 16965–16974.
32. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *39*, 1137–1149. [[CrossRef](#)]
33. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016*; Springer: Cham, Switzerland, 2016; pp. 21–37.
34. Kotthapalli, M.; Ravipati, D.; Bhatia, R. YOLOv1 to YOLOv11: A Comprehensive Survey of Real-Time Object Detection Innovations and Challenges. *arXiv* **2025**, arXiv:2508.02067. [[CrossRef](#)]
35. Panazan, C.E.; Dulf, E.H. Intelligent Cane for Assisting the Visually Impaired. *Technologies* **2024**, *12*, 75. [[CrossRef](#)]
36. Beyaz, A. Accuracy detection of intel® realsense D455 depth camera for agricultural applications. *Book* **2022**, 185–194.
37. Qiu, Z.; Lu, Y.; Qiu, Z. Review of ultrasonic ranging methods and their current challenges. *Micromachines* **2022**, *13*, 520. [[CrossRef](#)]
38. Papa, U.; Del Core, G.; Giordano, G.; Ponte, S. Obstacle detection and ranging sensor integration for a small unmanned aircraft system. In *Proceedings of the 2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), Padua, Italy, 21–23 June 2017*; IEEE: New York, NY, USA, 2017; pp. 571–577.

39. Li, N.; Ho, C.P.; Xue, J.; Lim, L.W.; Chen, G.; Fu, Y.H.; Lee, L.Y.T. A progress review on solid-state LiDAR and nanophotonics-based LiDAR sensors. *Laser Photon. Rev.* **2022**, *16*, 2100511. [[CrossRef](#)]
40. Patole, S.M.; Torlak, M.; Wang, D.; Ali, M. Automotive radars: A review of signal processing techniques. *IEEE Signal Process. Mag.* **2017**, *34*, 22–35. [[CrossRef](#)]
41. Abuelmakarem, H.; Abuelhaag, A.; Raafat, M.; Ayman, S. An Integrated IoT Smart Cane for the Blind and Visually Impaired Individuals. *SVU-Int. J. Eng. Sci. Appl.* **2024**, *5*, 71–78. [[CrossRef](#)]
42. Telkar, N.; Hiremath, P.; Harashetti, R. Vinetra Parijana: Smart Cane and Smart Glasses for Visually Impaired People. *May* **2023**, *3307*, 113–120.
43. Singh, K.; Vashisht, M.; Jyoti, D.S.; Tyagi, H. Navigation system for blind people using GPS & GSM Techniques. *Int. J. Sci. Res. Manag. Stud.* **2017**, *3*, 364–374.
44. Yousif, A.J.; Al-Jammas, M.H. A Lightweight Visual Understanding System for Enhanced Assistance to the Visually Impaired Using an Embedded Platform. *Diyala J. Eng. Sci.* **2024**, *17*, 146–162. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.