MJFAS
Malaysian Journal of Fundamental and Applied Sciences

RESEARCH ARTICLE

A Tensor Flow Lite-Powered Wearable Navigation Assistant Using Raspberry Pi for Real-Time Obstacle Detection and Autonomous Mobility in the Visually Impaired

Leong Kah Meng[a,c*], Ng Khai Le[a], Jahanzeb Sheikh[b,c], Ngeu Chee Hau @ Yeo Chee Hau[a], Tan Tian Swee[c], Kang Eng Siew[a], Chan Bun Seng[a], Chng Chern Wei[e], Jose-Javier Serrano Olmedo[d], Vasanthan A/L Maruthapillai[a]

[a]Faculty of Engineering & Information Technology, Southern University College, 81300 Johor, Malaysia; [b]Department of Biomedical Engineering, Sir Syed University of Engineering and Technology, Karachi, Pakistan; [c]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Johor, Malaysia; [d]Centre for Biomedical Technology Madrid, Universidad Politécnica de Madrid, Madrid, Spain; [e]Faculty of Computer Science and Technology, First City University College, Petaling Jaya, Selangor, Malaysia

Abstract In year 2023, around 2.2 billion people globally have near or distance visual impairment. Previous systems lacked comprehensive functionality, focusing only on basic obstacle detection or standalone features like fall detection. Moreover, the studies struggled with bulky designs, poor low-light performance, and limited environmental awareness. To address these challenges, the study develops ObstaSense, a wearable Electronic-Travel-Aid (ETA) for obstacle detection and navigation assistance. The system employed TensorFlow Lite, a Raspberry Pi-5, and a Pi-Camera Module-V3 to detect objects (e.g., people, potholes, vehicles) and relay avoidance instructions via Bluetooth earbuds. Its Real-Time Navigation (RTN) feature combined Global Positioning System (GPS), a compass sensor, and Plus Codes for precise guidance, enhanced by Google's Speech-To-Text (STT) and Text-to-Speech (TTS). Operating at 4–10 Frames Per Second (FPS), ObstaSense further integrated the Gemini Application programming interface (API) for multilingual (50-languages) image-to-text conversion. The system achieved consistent results by leveraging precise RTN functionality, which uses compass sensor data and vibration feedback to guide users accurately. Offline dataset training and evaluation were conducted solely to support the deployment of a real-time embedded assistive system on Raspberry Pi 5. Obstacle avoidance performance varied across rows, with the highest accuracy (100%) in the first row, followed by 66.7% in the third row and 50% in the second row. ObstaSense aids visually impaired, elderly, and cognitively impaired users, aligning with Sustainable Development Goals (SDGs) 3 and 10 for inclusive well-being.

Keywords: Camera Serial Interface; Electronic Travel Aid (ETA); Electronic Design Automation; Tensor processing unit; Text To Speech.

***For correspondence:**
kmleong@sc.edu.my

Received: 29 Sept. 2025
Accepted: 12 Jan. 2026

Introduction

With the development of more and more rapidly improving technology over the last couple of years, a very significant number of blind or visually impaired people are realizing that lots of the devices and apps today can assist them to live an almost entirely independent life [1]. Such technologies also include more advanced solutions, for instance, smart canes, e-solutions for traveling, GPS [2]. However, despite these advances, people are still in need of improved integrated, visually intuitive solutions that can help in real time in both dynamic and unfamiliar situations.

Age-related macular degeneration (26%) emerged as the primary cause of low vision while glaucoma (23%) followed closely behind and diabetic retinopathy (19%) ranked third according to affected individuals. Other posterior segment diseases (15%) and corneal opacity (7%) and complications from cataract surgery (4%) also contributed to low vision cases [3]. The European Council of Optometry and Optics defines low vision as an abnormal vision state which prevents normal execution of everyday activities. No treatment methods through optical correction or medical intervention can help this condition. The symptoms of vision disability typically include visual loss combined with reduced visual field as well as increased sensitivity to light and problems with color recognition and contrast sensitivity and night blindness. Blindness according to the International Classification of Diseases (ICD-10) exists when a person has vision less than 0.05 (20/400, 3/60, 1.3 logMAR) or the corresponding visual field loss in the best-corrected eye measures less than 10 degrees. People with visual impairments have difficulty recognizing objects [4]. The utilization of a traditional white cane, guide dogs, and mobility training would be included in the range of expertise required to operate in the field of orientation and mobility to assist visually impaired individuals. A typical white cane has a restricted detection range of obstacles, dependent on the cane's length. As a result, this reduces the user's walking speed. Moreover, recent research in the area of visually impaired presented by [5] have put emphasis on assistive technologies especially in the areas of object detection and navigation. Their approach was well-suited for embedded systems due to its low computational requirements. Nevertheless, their work did not include features like fall detection, emergency response, or communication support, which are vital in real-life assistive scenarios. Others include the traditional devices which characteristically include sensor-based devices such as the laser or the ultrasonic systems which tend to limit the information given to users and cause unsafe navigation. The literature suggests a more sophisticated system that employs what is known as Convolutional Neural Network (CNN) based object detecting and tracking system with special focus on MobileNet architecture that provides real time detection of objects while having low power consumption. Traditional techniques (such as the Viola-Jones framework) have proved effective in face detection, but have found wider application in tasks such as detecting cars and pedestrians. With the introduction of CNN and GPU-accelerated deep learning frameworks, algorithms such as Overfeat, R-CNN, Fast R-CNN, Faster R-CNN, R-FCN, SSD, and YOLO set new performance standards that significantly improve the accuracy and efficiency of object detection systems. The knowledge from object detection methods helps in choosing the system architecture and the right algorithms to ensure high efficiency and accuracy in the real applications [6].

Several previous studies have explored assistive technologies to support visually impaired individuals, focusing on object detection and navigation. A navigation tool is crucial for guiding users along paths and helping them perceive their surroundings by superimposing digital information onto the real world. Object recognition plays a vital role in this technology, serving as a key component in advanced assistive systems for the visually impaired. For such purpose, Konaite and fellows [7] developed a real-time object detection system using Raspberry Pi 4 and the SSD MobileNet V2 model. Their goal was to help users identify road signs and obstacles. While the system achieved satisfactory performance in object recognition, it was mainly tested in static outdoor environments. The lack of dynamic obstacle handling and additional safety features such as fall detection limited its practicality in real-world scenarios. Although the existing system improved overall user interaction, it is relatively bulky and not fully optimized for compact, wearable use, which can reduce comfort and convenience. Another study [8] focused on edge detection by using the Histogram of Oriented Gradients (HOG) method. Their system showed promising results, especially for tasks like face and basic object detection. However, their implementation was not tailored for visually impaired users and lacked practical features such as environmental awareness or indoor usability. Similarly, [9] introduced a wearable system using an ESP32 microcontroller, ultrasonic and PIR sensors, an accelerometer, and Bluetooth communication. Their solution was capable of detecting obstacles and falls with high accuracy (98%). It also featured GPS-based location tracking and emergency messaging. Despite this, the system lacked computer vision capabilities and indoor navigation functions, which could limit user independence in complex environments.

Thus, a navigation system is crucial for visually impaired individuals daily activities since it helps them avoid obstacles and provides precise directions to their destination [10]. The challenge is to develop a navigation system that allows visually impaired people to be navigated through indoor and/or outdoor landscapes and terrains, encouraging a sense of security when moving in unfamiliar environments. Most of the currently available ETA's obstacle detection capability for the visually impaired individuals are insufficient of providing the detailed representation of the territory [11, 12]. As a result of these performance problems and the high costs of complex electronic aids, the great majority of visually impaired depend on the use of the much cheaper long cane [13]. Thus, there is a clear need for an affordable, effective and easily accessible navigation in order to avoid obstacles, and immediately and continuously give feedback without blocking other senses. The ObstaSense is suitable for fully blind, low vision, and visually impaired individuals, as well as people with cognitive or amnesia disabilities. Featuring obstacle avoidance system, and RTN with voice assistant, the ObstaSense aligns well with the United Nations SDG 3 and 10 by 2030 Agenda for Sustainable Development: Good Health and Well-Being, and Reduced Inequalities. According to the 2023 survey of World Health Organization (WHO) about 2.2 billion of people have near or distance vision impairment [14, 15]. The ObstaSense promotes healthy lives and well-being by improving physical safety and mental health through independent navigation [16 – 18]. These statistics highlight the need of ETA to address the visually impaired on a international scale. When faced with unfamiliar surroundings or terrains, visually impaired people have difficulty navigating to their desired destinations and can feel insecure without pedestrian tactile facilities [19 – 21].

To overcome and cater the present challenges, the study proposes an ETA called The ObstaSense, which features an obstacle avoidance technology with Tensorflow 2 object detection API and RTN with voice assistant for enhancing navigation experience. RTN in contrast, is the second innovative features integrated in the ObstaSense. The RTN feature allowed the tracking of navigation path in the real world via camera and provided users with real-time navigational instructions through Bluetooth earbuds. Furthermore, it helped in independence because the technology enabled the users to move from one place to the other without subjecting them to risk of life. The RTN also included a voice assistant that users can interact with by speaking directly to the Bluetooth earbuds’ microphone through voice commands. This voice assistant provided immediate answers and being equipped with TTS and STT capabilities. Furthermore, the RTN, which uses a GPS sensor, compass sensor, and vibration motor for haptic feedback, guides the user from A to B, node to node, with directional instructions. This allows the user to walk to the destination in the right direction, reducing circulating around the destination. The primary objective of this study is to design, implement, and evaluate a real-time wearable ETA for visually impaired users. Offline model training and image analysis are performed only as preparatory steps to enable efficient real-time deployment on embedded hardware.

Materials and Methods

Overview of the Study

The ObstaSense is a novel ETA developed to enhance mobility and independence for individuals with visual impairments. The ObstaSense is a novel ETA developed to enhance mobility and independence for individuals with visual impairments. It integrates real-time object detection using the TensorFlow Lite Object Detection API and a RTN system powered by a voice assistant. The system overlays digital information on the physical environment to provide directional cues and obstacle detection. A dataset of labeled images of obstacles is used to train the SSD MobileNet V2 model, which is deployed on a Raspberry Pi 5. The Raspberry Pi Camera Module V3 captures real-time images for processing, and the model continuously adapts with new real-world data to enhance accuracy. The RTN system employs Bluetooth earbuds for voice instructions and incorporates STT and TTS functions, allowing users to issue voice commands and receive audio feedback. It also utilizes OpenRouteService APIs for intelligent location-based navigation, even when only partial destination names are provided.

Study Workflow

The development of ObstaSense was carried out in two structured phases: Phase 1 and Phase 2 as shown in Figure 2.1. Phase 1 focused on the foundational groundwork, beginning with the discussion of the project title, a comprehensive literature review, and a market survey aimed at understanding user needs and expectations. Following this, the core technologies were introduced, including the development and testing of TTS, STT, and object detection using the TensorFlow Object Detection API. Preliminary features for passive navigation were integrated, and early results were collected to evaluate system feasibility. Phase 2 marked a major advancement with the full integration of a voice assistant and image processing components under the RTN framework. This phase also included the complete development of obstacle avoidance algorithms and haptic feedback mechanisms, resulting in a more robust and interactive user interface. The entire system was integrated and subjected to extensive prototyping, testing, and optimization, culminating in the design and fabrication of a custom 3D-printed enclosure, as well as the execution of field experiments to validate real-world usability and accuracy. This real-time inference strategy illustrates the sequential logic and safety checks integrated into the obstacle avoidance module.

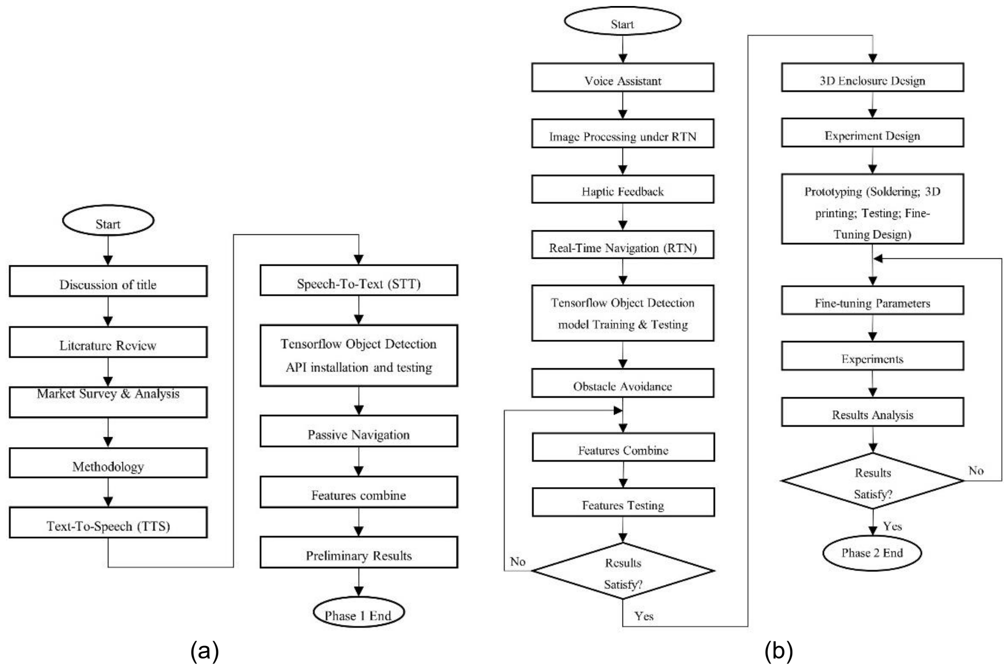


Figure 2.1. Full Project Development Workflow for; (a) Phase 1 and (b) Phase 2

Obstacle Avoidance

The obstacle avoidance system is a core feature of ObstaSense, enabling visually impaired users to detect and navigate around physical barriers in real-time. This feature integrates computer vision, machine learning, and embedded hardware to interpret surroundings and guide user movement through intuitive feedback.

Raspberry Pi 5

At the heart of the system is the Raspberry Pi 5, chosen for its high processing capability and low-power profile. It features a quad-core 1.8GHz ARM Cortex-A76 processor and supports up to 8GB LPDDR4 RAM, ensuring smooth multitasking and efficient model inference. It includes essential communication interfaces such as dual-band Wi-Fi, Bluetooth, and Gigabit Ethernet, and supports Raspberry Pi OS. Though equipped with both CPU and GPU, the system prioritizes CPU power for real-time data processing, image handling, and navigation logic.

Raspberry Pi Camera Module V3

The Camera Module V3 is used to capture high-resolution input images for object detection. It employs a Sony IMX708 12MP sensor and offers a 120° wide field-of-view, supporting video capture up to 1080p @ 30 FPS. Notably, it includes NoIR (no infrared filter), improving low-light and night-time performance. It connects via the MIPI CSI interface for high-speed data transfer and features Phase Detect Autofocus (PDAF) for image clarity without user intervention.

IR LED Module

To enhance night vision, two 3W Infrared (IR) Light-Emitting Diodes (LEDs) are integrated with the NoIR camera. The IR LEDs activate automatically under low-light conditions, triggered by a built-in Light Dependent Resistor (LDR). A mini potentiometer allows threshold adjustment. These IR modules emit invisible light, making them ideal for discrete, non-distracting navigation aids.

Implementation of TensorFlow

Table 2.1 summarizes the object detection development workflow in the ObstaSense system, highlighting the transition from model training in TensorFlow 2 to real-time deployment using TensorFlow Lite. It outlines key processes such as image labelling, model training, fine-tuning, and conversion to the TFLite format. Due to the limited memory and computational resources of embedded platforms such as the Raspberry Pi 5, model size and runtime efficiency were key considerations in the system design.

TensorFlow Lite was adopted to reduce memory footprint and enable real-time inference while maintaining acceptable detection accuracy. The table also details how default COCO models are optimized and combined with custom-trained models to enhance detection accuracy while maintaining performance on the Raspberry Pi 5. This structured pipeline enables reliable and efficient obstacle detection suitable for real-time assistive applications.

Table 2.1 Summary of TensorFlow 2 and TensorFlow Lite Implementation in ObstaSense

Section	Process	Description
TensorFlow 2	Image Labelling	Manual annotation using a Protobuf-based tool to assign class labels and bounding boxes, including custom objects not in the COCO dataset.
	Image Training	Training the SSD MobileNet V2 model using the annotated dataset, allowing the model to learn object classification and localization.
	Evaluation & Fine-Tuning	Performance tested on a validation dataset; hyperparameters adjusted to reduce overfitting and improve accuracy.
	Model Conversion	Trained model converted to TensorFlow Lite (.tflite) format to optimize it for edge deployment on Raspberry Pi 5.
TensorFlow Lite	Model Preparation	TensorFlow Lite Converter used to shrink the model size while retaining detection accuracy for edge device inference.
	Default Model Optimization	COCO-pretrained TFLite model pruned to retain relevant object classes (e.g., person, chair) for efficiency.
	Custom & Default Model Fusion	Custom model combined with optimized COCO model to support both general and specific detection requirements.
	Deployment	Combined TFLite model deployed on Raspberry Pi 5 using TensorFlow Lite runtime for real-time object detection.
	Inference & Testing	Real-time camera inputs processed; object locations and labels generated for navigation decisions.

As shown in Figure 2.3b, the region design of the algorithm pays attention to different zones, thereby increasing the detection accuracy. Moreover, Figure 2.3c presents the flowchart of the obstacle avoidance algorithm, detailing the real-time decision-making process that enables ObstaSense to guide users safely. The system begins by initializing key components, including the TensorFlow Lite model and the camera interface. Each captured frame is preprocessed and analyzed to detect objects using the trained model. The image is segmented into three regions, left, center, and right, and the available space in each is assessed based on detected obstacles. Depending on the location and proximity of these objects, the system generates navigation commands such as “go straight”, “turn left”, “turn right”, or “stop”. If an object is detected within 100 pixels, the system prompts a stop or reverse action to prevent collisions. The entire process runs in a loop until the user terminates it, ensuring continuous, adaptive navigation support. This spatial segmentation enables the system to localize objects within specific areas of the user's field of view, enhancing the granularity of the detection process. By independently evaluating each region, the system minimizes the likelihood of overlapping or ambiguous detections, which could compromise accuracy. This structure allows ObstaSense to make context-aware navigation decisions, such as turning left or right, based on the availability of space in each region. As a result, the system can offer more precise and reliable guidance, especially in dynamic or cluttered environments

Real-Time Navigation (RTN) System

The RTN system enhances ObstaSense by offering intelligent, voice-controlled destination guidance for visually impaired users. It integrates a voice assistant powered by STT and TTS technologies, enabling hands-free interaction through Bluetooth earbuds. Users can issue partial or complete destination names, which are processed using OpenRouteService APIs (Geocode Autocomplete, Reverse Geocode, and Directions) to retrieve coordinates and calculate optimal walking paths. The passive navigation phase fetches real-time GPS coordinates, while image processing concurrently analyzes the environment to avoid obstacles. The system provides vibration feedback using motors to indicate heading direction, guided by compass bearing calculations derived from the QMC5883L compass sensor and MPU6050 IMU sensor. This multimodal feedback (voice + haptics) ensures users stay aligned with their intended path, even in unfamiliar environments.

Hardware Architecture and System Integration

The hardware architecture of ObstaSense consists of a Raspberry Pi 5 as the central processing unit, interfaced with a Pi Camera Module V3 for visual input, infrared LED modules for low-light illumination, GPS and compass sensors for navigation, vibration motors for haptic feedback, and Bluetooth earbuds for audio interaction. All components are powered by a portable battery supply and integrated within a chest-mounted wearable enclosure to enable hands-free operation. A block diagram illustrating the interconnection and data flow between hardware components is provided to support the system description. Figure 2.2 Hardware block diagram of the ObstaSense system showing parallel sensor inputs, Raspberry Pi 5–based processing with TensorFlow Lite, and multimodal user feedback through audio and haptic outputs.

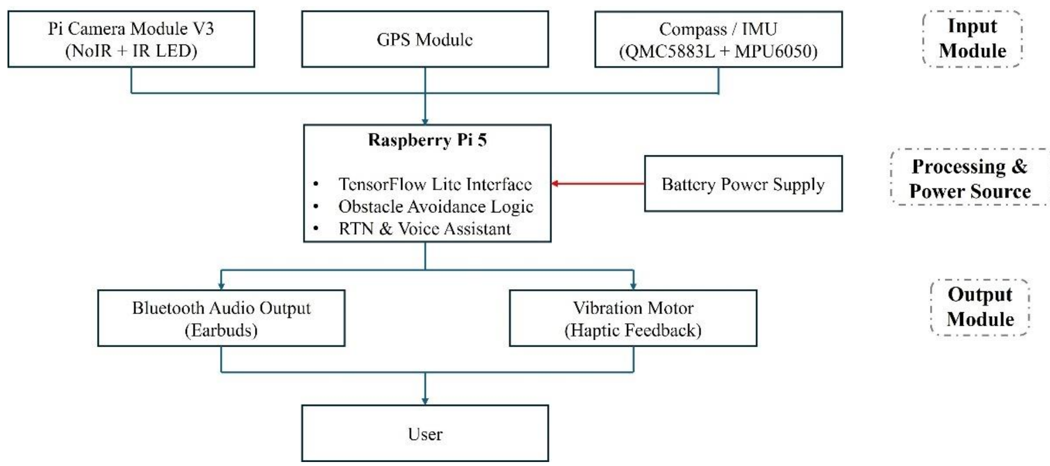


Figure 2.2 Hardware architecture and system integration of ObstaSense

Prototype Development

The final prototype (Figure 2.3a) was developed through iterative mechanical and electronic integration. Five design structures were evaluated for wearability, stability, and user discretion. The selected Design Structure 5, a chest-mounted, 3D-printed enclosure—balances functionality and aesthetics, and includes braille markings, metal push buttons, and secure stitching holes for chest bag mounting. The enclosure was designed in Autodesk Fusion 360, 3D-printed with PLA, and hand-sewn to a chest bag for comfort and social acceptability. The prototype incorporates sensors, GPS, camera, IR LEDs, and computing components, all wired as per the KiCad-generated schematic diagram. The total device weight is ~650g, with weight distribution ensuring user comfort during extended use

Experimental Setup

There are three distinct regions used to dividing the captured image into left, center, and right rectangular boxes. These rectangular regions are essential for localizing and focusing object detection tasks within specific areas of interest. The spatial independence is achieved which includes the left region, the center region and the right region, where these regions strategically cover the entire frame. By separating the evaluation of each segment, this method reduces the chance of overlapping detections that degrade the overall accuracy of the detection method. The system can work more precisely because it can independently analyze each region and adapt to real time conditions better. As shown in Figure 2.3b, the region design of the algorithm pays attention to different zones, thereby increasing the detection accuracy.

For the experiment (Figure 2.3c), two experiments were conducted to validate the system. The Obstacle Avoidance Test involved a pyramid-shaped obstacle layout using chairs, where blindfolded participants relied on ObstaSense's auditory cues to navigate safely. This assessed detection accuracy and decision reliability as obstacle density increased. The Navigation Experiment required participants to walk from one campus location to another using the RTN system, guided only by vibration and voice instructions while blindfolded. The experiment evaluated the system's real-world usability, path accuracy, and user safety. Both experiments confirmed ObstaSense's functionality and highlighted its potential as a reliable mobility aid.

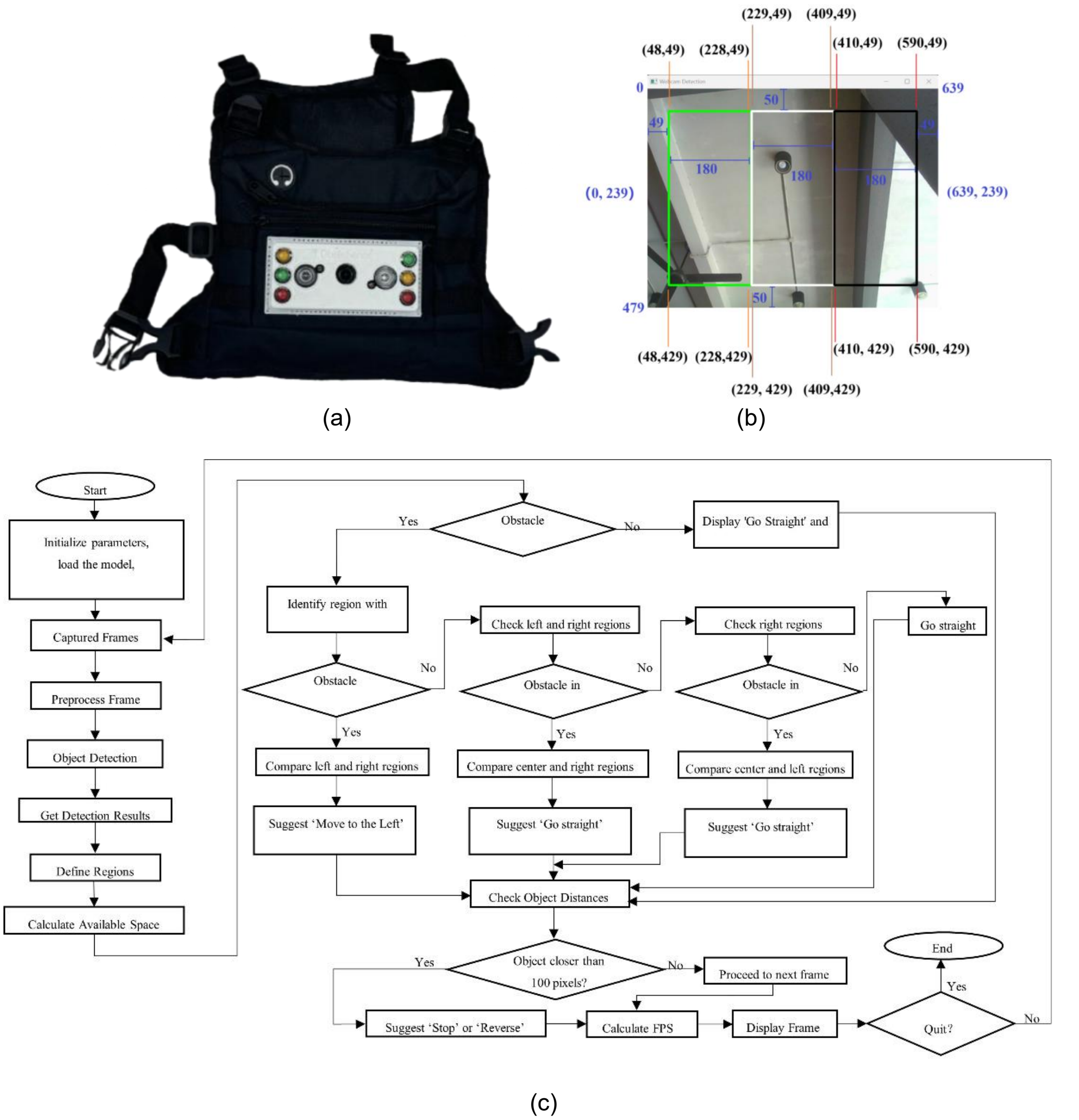


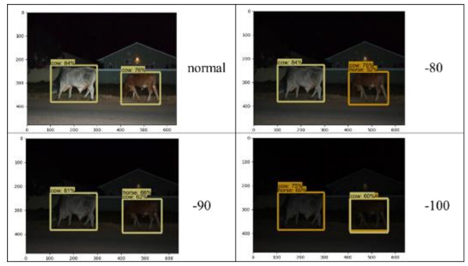
Figure 2.3 Overall flow of experimental setup (a) Prototype Design, (b) Region design of Algorithm, (c) Flowchart of obstacle avoidance system

Results and Discussion

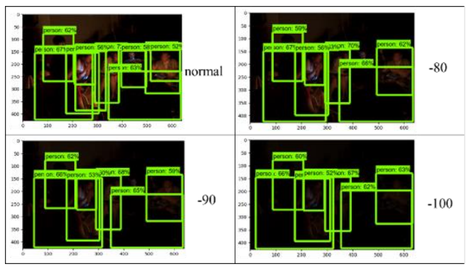
Under standard indoor and outdoor lighting conditions, the camera module achieved stable object detection performance, providing a baseline reference for subsequent low-light evaluations. This baseline performance allows the impact of brightness reduction and infrared assistance to be assessed more clearly.

Object Detection Implementation for Modelling in Low Light Performance

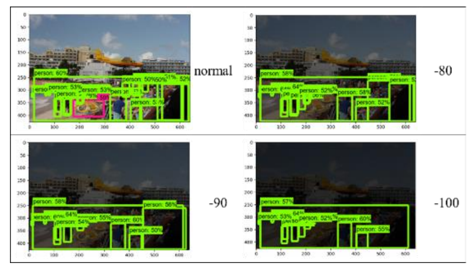
The system demonstrated robustness in detecting objects under low-light conditions, though with some limitations. At 80% brightness reduction, the model occasionally misclassified cows as horses (Figure 3.1a). However, person detection accuracy remained stable in dark indoor environments, albeit with a reduced field of view in settings like KTV (Figure 3.1b). In normal outdoor lighting, object detection was efficient, but low-light conditions restricted detection primarily to people, with a slight reduction in range (Figure 3.1c). These findings highlight the model's potential for moderate darkness but underscore the need for further optimization in extreme low-light scenarios.



(a)



(b)

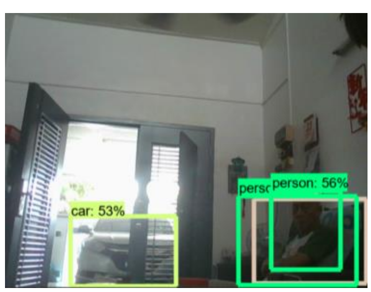


(c)

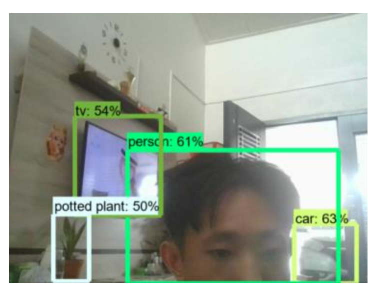
Figure 3.1 Examples of brightness reduction results under different scenarios: (a) The result of brightness reduction in the model between a cow and a horse, (b) The result of brightness reduction in KTV, and (c) The result of brightness reduction in normal lighting condition

Real-Time Multi-Object Detection using SSD MobileNet V2-Model

The SSD MobileNet V2 model successfully performed real-time multi-object detection, identifying objects such as persons, cars, and potted plants (Figure 3.2). This capability is critical for providing comprehensive environmental feedback to visually impaired users, ensuring safer navigation in dynamic settings. Such tests were first performed on a laptop in order to study the efficiency of the SSD model before its application in Raspberry Pi 5. As seen in Figure 3.2a and Figure 3.2b, actual result of the objects like person, car and potted plant are detected. This is a good example of how the model is capable of differentiating multiple objects in a scene as it plays a great role in giving environmental feedback to users with impaired vision.



(a)



(b)

Figure 3.2 Real-Time Multi-Object Detection (a) Multiple Detections, including a Car and a Person (b) Multiple Detections, including a TV, Potted Plant, Car, and Person.

Training and Dataset Improvements

Initial training for manhole cover detection yielded poor accuracy due to limited and imbalanced datasets (Figure 3.3a). Manhole cover detection was performed during the initial training using TensorFlow 2 following a dataset of around 150 images taken from online channels. The images were labelled and converted from XML to CSV format and 80% for training and 20% for testing. It was about 12 hours of training. Nevertheless, the model achieved poor accuracy, as objects were often incorrectly and correctly marked as “manhole cover”. Subsequent iterations introduced lamp poles and drains, but accuracy remained suboptimal for the following detection classes which were added for lamp, poles and drains. A dataset of 450 images was prepared where 200 images were of lamp poles and 250 for drains. Figure 3.3b shows results after labelling and training the model, which are still unsatisfactory given that many objects were incorrectly labelled as “drain”, like the previous results. Balancing the dataset with high-quality images and equal class distribution improved detection slightly (Figure 3.3c). To improve upon the original experiments the dataset was edited in order that some of the lower quality images were replaced with new sets of high-quality photos brought into the study, giving a set of 600 images. Images of lamp poles and drains were balanced with equal distribution to the dataset. This small adjustment increased detection accuracy slightly, as the model began to distinguish between objects better. Although, accuracy was still too suboptimal and in real time the model often misclassified objects while correctly predicting labels without any objects, this demonstrated that achieving high reliability witness these situations remains a large challenge. In addition, the limited number and the imbalanced images among different classes contributed to the low accuracy.

To enhance the performance of the object detection model, significant improvements were made to the training dataset and methodology. Low-quality images sourced from the internet were replaced with over 500 high-quality, manually collected images, ensuring better feature representation. The labeling process was refined by excluding light components from poles and focusing solely on the pole structure, which improved detection accuracy for chest-level wearable applications (Figure 3.3d). Additionally, segmented labelling was employed to minimize irrelevant background elements, further enhancing precision. To increase dataset diversity, data augmentation techniques such as random horizontal flips, 90-degree rotations, SSD random cropping, and pixel value scaling were applied, significantly improving model robustness. Training optimization included a cosine decay learning rate strategy (base rate: 0.05 over 20K steps, warmup rate: 0.02 for the first 1K steps), ensuring stable adaptation during training. Despite these improvements, the custom-trained model's effectiveness remained suboptimal when deployed on the Raspberry Pi 5 due to hardware limitations in handling complex models (Figure 3.3e). Achieving higher accuracy would require a larger, more diverse dataset and extended training sessions, which were beyond the project's scope. These findings highlight the trade-offs between model complexity and real-time performance on embedded systems.

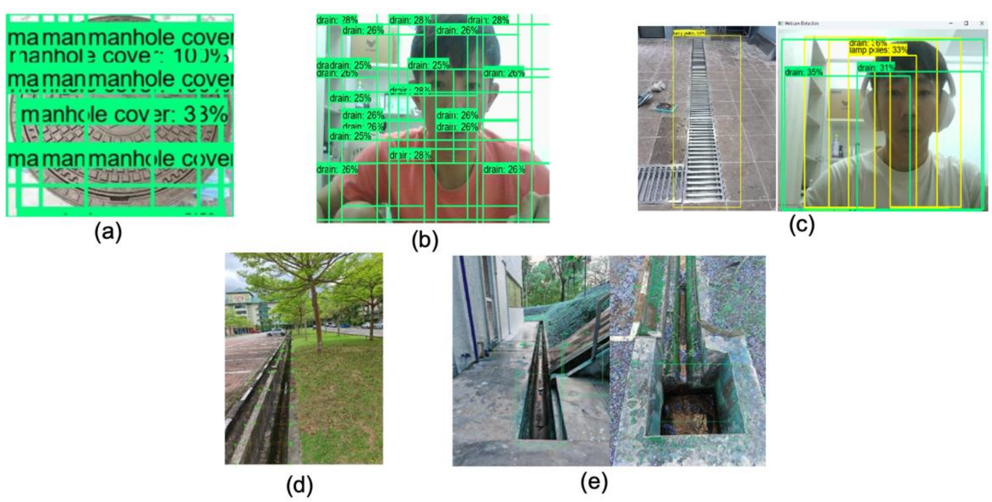


Figure 3.3 Evolution of object detection performance: (a) Initial poor accuracy in manhole cover detection using 150 images; (b) Continued misclassification issues after expanding to lamp poles and drains (450 images); (c) Slight improvement with balanced high-quality dataset (600 images); (d) Enhanced pole detection through refined labeling of manually collected images; (e) Final suboptimal performance on Raspberry Pi 5 despite data augmentation and learning rate optimization

NoIR Camera Performance, Object Detection and Avoidance

The NoIR Camera Module 3, devoid of an IR filter, captured high-resolution images with a pink hue under certain lighting. Adjusting camera parameters particularly exposure and gain improved low-light image quality, though overexposure and noise were trade-offs. Integrating

IR light significantly enhanced visibility in complete darkness, making it ideal for nighttime navigation. Moreover, IR illumination enabled object detection in dark environments, though accuracy was compromised due to limited resolution and detail. Without IR light, no objects were detected, emphasizing IR's critical role in low-light navigation.

The system divided the environment into left, center, and right regions to guide users around obstacles (Figures 3.4a). For multi-object scenarios, users were directed toward the least obstructed region (Figure 3.4b). A proximity based “Stop or Reverse” warning was triggered for nearby objects, enhancing safety.

Figure 3.4 NoIR Camera Module 3 Performance in Low-Light Navigation (a) Environment divided into left, center, and right regions for obstacle avoidance. (b) Multi-object navigation directing users toward the least obstructed path.

The experiment further evaluates object detection accuracy at certain specified distances using a tape measure for verification. In Figure 3.5a, one meter of distance is taped and verified by a tape measure using a system which successfully detects the object. In Figure 3.5b, both distance and detection accuracy remain accurate using a 5-meter tape measure to measure a 2-meter distance, and in Figure 3.5c distance is again measured using a 5-meter tape. In its absence of obstacles, Figure 3.5c, Figure 3.5d and Figure 3.5e showed that the system gave consistent detection accuracy for distances up to 5 meters, thereby proving its validity for clear weather. But the detection accuracy for vehicles decreases as obstacles are incident, as shown in the Figure 3.5b, Figure 3.5c and Figure 3.5d. When objects are nearby, the system has difficulty detecting accurately the distance to the cars when they are closer or farther away. Figure 3.5f features two individuals and one car, but the detection system is unable to distinguish the closer car because of the lack of accuracy in a complex environment, full of obstacles, with overlapping objects. As a result, these observations imply that for detecting in challenging scenarios with multiple and overlapping objects, the algorithm will benefit from optimization.

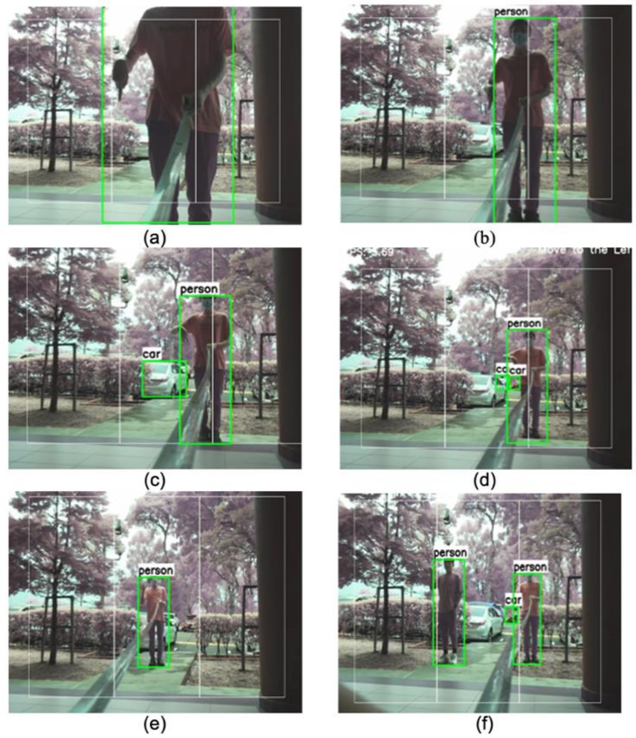


Figure 3.5 Accuracy of Object Detection at Different Distances (a) Object Detection at 1 Meter (b) Object Detection at 2 Meters (c) Object Detection at 3 Meters (d) Object Detection at 4 Meters (e) Object Detection at 5 Meters (f) Multi-Person Vertical Detection at 5 Meters

Detection Accuracy

The performance of the obstacle avoidance system under varying numbers of obstacles (chairs) across three spatial rows was tested with six users in each configuration. The system demonstrated perfect accuracy (100%) when 1 to 3 chairs were present in the first row, with all six users avoiding obstacles without any collisions. This suggested high reliability in low-complexity environments where obstacles are minimal and restricted to a single plane. However, as the number of chairs increased and were distributed across multiple rows, system performance declined. When a fourth chair was introduced into the second row, accuracy dropped to 50%, with three users failing to navigate the path without incident. Collision data indicated users most hit obstacles located in the second row, with one user hitting an object twice and two others colliding once.

The situation further deteriorated with five chairs. The accuracy declined sharply to 33.33%, with only two users navigating successfully. Multiple collisions were recorded, including a user hitting obstacles in both the first and second rows. The second row consistently emerged as a critical zone, accounting for most collisions across increasing obstacle density. Interestingly, with seven chairs placed across all three rows, the accuracy slightly improved to 66.66%, suggesting that the users adapted better or that spatial distribution allowed for improved navigation paths. Even then, collisions still occurred across all rows, particularly the second and third rows, confirming their higher difficulty level for obstacle detection and avoidance. Overall, the data indicates that while the system is highly effective in simple layouts, its performance diminished in more complex and densely populated environments, especially when obstacles appear beyond the user's immediate forward view. These findings underscored the importance of enhancing mid- and peripheral-range sensing, especially for multi-row obstacle environments.

Navigation and Battery Performance

The results in Table 3.1 demonstrate how varying search radius and result size parameters impact geolocation accuracy and system responsiveness. The Geocode Autocomplete API was able to effectively identify nearby destinations within 0.5 km and 1 km radii, delivering identical outputs across all sizes (5, 10, and 15). This indicates that increasing the result size does not enhance accuracy within these smaller radii, thus conserving computational resources and supporting energy-efficient operation. Optimizing memory usage and computational load also contributed to stable system operation and energy efficiency during prolonged real-time navigation.

However, when the radius was increased to 5 km, the number of results expanded considerably, and the location distances grew significantly. For example, Entry 7 under the 5 km radius returned a result as far as 4.779 km, which is impractical for short-range, pedestrian-based navigation, especially for visually impaired users. These distant results not only complicate routing decisions but also extend navigation times and increase battery consumption in wearable or mobile systems. Importantly, the consistent presence of values across entries 1–5 for all sizes and radii validates the robustness of the API for core destinations. But starting from Entry 6 onward, results only appeared under larger radii, highlighting the limited discovery potential within smaller radii unless the size of results is expanded. Combining this with performance data from the OpenRouteService API, which consistently detected a maximum of 8 valid "bus" destinations, the analysis confirms a 1 km radius as the optimal trade-off between performance and practicality. This boundary ensures reliable location capture while avoiding the inclusion of irrelevant distant points that strain battery life and complicate path planning. Ultimately, a search size of 10 with a 1 km radius offers a balanced configuration, enabling precise geolocation (refined using Plus Codes to within 2.8 m × 3.5 m) without compromising system efficiency or navigation reliability.

Table 3.1 Geocode Autocomplete API Search Results; Size = 5, 10 and 15

	Size (5)			Size (10)			Size (15)		
No	Radius (km)			Radius (km)			Radius (km)		
	0.5	1	5	0.5	1	5	0.5	1	5
1	0.872	0.872	0.872	0.872	0.872	0.872	0.872	0.872	0.872
2	0.2	0.2	1.073	0.2	0.2	1.073	0.2	0.2	1.073
3	0.251	0.251	1.222	0.251	0.251	1.222	0.251	0.251	1.222
4	0.308	0.308	1.31	0.308	0.308	1.31	0.308	0.308	1.31
5	0.456	0.456	1.486	0.456	0.456	1.486	0.456	0.456	1.486
6	-	-	-	0.498	0.498	1.508	0.498	0.498	1.508
7	-	-	-	0.599	0.599	4.779	0.599	0.599	4.779
8	-	-	-	0.787	0.787	0.2	0.787	0.787	0.2
9	-	-	-	-	-	0.251	-	-	0.251
10	-	-	-	-	-	0.308	-	-	0.308
11	-	-	-	-	-	-	-	-	0.456
12	-	-	-	-	-	-	-	-	0.498
13	-	-	-	-	-	-	-	-	0.599
14	-	-	-	-	-	-	-	-	0.787
15	-	-	-	-	-	-	-	-	1.143

Comparison of ObstaSense with Existing Assistive Systems

Compared to prior works, our ObstaSense introduced several novel components and improvements. While existing systems, such as those by Konaite *et al*. [7] and Ashiq *et al*. [5], primarily focused on object detection and obstacle avoidance, our project extends functionality with emergency communication, fall detection, and indoor navigation. Unlike Ghosh *et al*. [2], who integrated emotion detection and Artificial Intelligence (AI) audio guidance in a bulky format, our design maintains a compact form factor in the shape of a chest bag. Andrea [8] focused on face recognition using HOG, but lacked contextual environment understanding. Moreover, Rahman [9] provided strong fall detection but no advanced machine vision or light adaptation. In contrast to all the studies, our system incorporated IR light to operate effectively in low-light conditions, and integrates smartphone communication, emergency recording, wall detection, and indoor navigation to form a comprehensive solution. The ObstaSense project stood out due to its holistic and user-focused design tailored for real-world application. One of the key novelties is the integration of the system into a chest bag, ensuring discreet usage and reducing stigma often associated with assistive devices. Unlike traditional handheld tools or bulky hardware, the chest-mounted form allowed hands-free operation while maintaining aesthetics. Overall, the IR enables effective object detection and navigation in low-light environments, something most previous systems lacked. Additionally, the device combined multiple vital safety features such as fall detection, wall detection, indoor navigation, emergency voice recording, and even direct phone calling in one compact package. This all-in-one solution provides not only mobility support but also enhances the safety and autonomy of the visually impaired. A detailed comparison of ObstaSense with conventionally available and other recently developed obstacle detection systems is presented in Table 3.2. The table evaluates key parameters such as detection accuracy, response time, multi-obstacle recognition, adaptability to environmental conditions, power efficiency, and cost-effectiveness.

Table 3.2 Comparative Analysis of ObstaSense with Existing Systems

Feature	ObstaSense	Konaite *et al.*	Ghosh *et al.*	Andrea *et al.*	Ashiq *et al.*	Rahman *et al.*	Traditional Products
Object Detection	✔	✔	✔	✔	✔	NA	NA
Fall Detection	✔	NA	NA	NA	NA	✔	NA
IR Light (Night Vision)	✔	NA	NA	NA	NA	NA	NA
Indoor Navigation	✔	NA	NA	NA	NA	NA	NA
Wall Detection	✔	NA	NA	NA	NA	NA	NA
Emergency Recording	✔	NA	NA	NA	NA	NA	NA
Phone Calling	✔	NA	NA	NA	NA	✔	NA
Location Tracking	✔ via GPS	NA	NA	NA	NA	✔	NA
Wearable (Chest Bag)	✔	NA	NA	NA	NA	✔	NA

Conclusions

The ObstaSense ETA has been successfully developed and tested as a functional wearable device designed to enhance mobility for visually impaired individuals, the elderly, and those with cognitive disabilities. Combining obstacle avoidance, real-time navigation, voice assistance, and haptic feedback, the system provides a comprehensive solution that promotes independence and safety. Performance evaluations confirmed its effectiveness in detecting single obstacles with 100% accuracy, though improvements are needed for multiple obstacle scenarios. The integration of Plus Codes significantly enhanced navigation precision by simplifying destination targeting, while voice commands and vibration feedback ensured a user-friendly experience. Additionally, the device demonstrated partial functionality in low-light conditions using IR LEDs, proving viable in controlled tests. However, limitations such as GPS cold-start delays, indoor navigation challenges, and a limited battery life of four hours highlight areas for future refinement. Addressing these through higher-capacity batteries, optimized power consumption, and faster GPS modules will further enhance the device's reliability. Overall, the ObstaSense represents a significant step forward in assistive technology, aligning with global sustainability goals (SDGs 3 and 10) and supporting Malaysia's Industry 4.0 vision. Continued development will focus on improving real-time performance, obstacle detection, and user comfort to ensure broader accessibility and adoption.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgment

The authors would like to thank the Southern University College Research Fund (SUCRF/C2-2024/FEIT-23) for financially supporting this work and Sir Syed University of Engineering and Technology for the research and technical support provided to polish the research.

References

[1] Patel, I., Kulkarni, M., & Mehendale, N. (2024). Review of sensor-driven assistive device technologies for enhancing navigation for the visually impaired. *Multimedia Tools and Applications, 83*, 52171–52195.

[2] Ghosh, A., Mahmud, S. A., Uday, T. I. R., & Farid, D. M. (2020). Assistive technology for visually impaired using TensorFlow object detection in Raspberry Pi and Coral USB accelerator. In *IEEE Region 10 Symposium (TENSYMP)*.

[3] Cardona, A. A., & Vasquez, S. (2021). Mobility aids for visually impaired persons: Journals reviewed. *Wearable Technology, 1*, 73–81.

[4] Nazri, N. M., Fauzi, S., Gining, R., Razak, T. R., & Jamaluddin, M. (2021). Smart cane for visually impaired with obstacle, water detection and GPS. *International Journal of Computing and Digital Systems, 10*.

[5] Ashiq, F., Asif, M., Ahmad, M. B., Zafar, S., Masood, K., & Mahmood, T. (2022). CNN-based object recognition and tracking system to assist visually impaired people. *IEEE Access, 10*, 14819–14834.

[6] Padilla, R., Netto, S. L., & Da Silva, E. A. (2020). A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*.

[7] Konaite, M., Owolawi, P. A., Mapayi, T., Malele, V., Odeyemi, K., & Aiyetoro, G. (2021). Smart hat for the blind with real-time object detection using Raspberry Pi and TensorFlow Lite. In *Proceedings of the International Conference on Artificial Intelligence and its Applications*.

[8] Andrea, R., Ikhsan, N., & Sudirman, Z. (2022). Face recognition using histogram of oriented gradients with TensorFlow in surveillance camera on Raspberry Pi. *International Journal of Information Engineering & Electronic Business, 14*(1). https://doi.org/10.5815/ijieeb.2022.01.01.

[9] Rahman, M., Islam, M., Ahmed, S., & Khan, S. A. (2020). Obstacle and fall detection to guide the visually impaired people with real time monitoring. *SN Computer Science, 4*, 219. https://doi.org/10.1007/s42979-020-00219-1.

[10] Okolo, G. I., Althobaiti, T., & Ramzan, N. (2024). Assistive systems for visually impaired persons: Challenges and opportunities for navigation assistance. *Sensors, 24*, 3572. https://doi.org/10.3390/s24093572.

[11] Bujacz, M., & Strumiłło, P. (2016). Sonification: Review of auditory display solutions in electronic travel aids for the blind. *Archives of Acoustics, 41*, 401–414. https://doi.org/10.1515/aoa-2016-0042.

[12] Patil, L. H., Dahale, N. P., Thakare, A. L., Fursule, S. S., & Vaidya, A. V. (2024). Visionary assistance: Integrating ultrasonic and GPS technologies for the visually impaired. *International Journal of Innovative Research in Technology and Science, 122*, 410–415. https://doi.org/10.1234/ijirts.2024.122.410.

[13] Bismark, A. A. (2024). *Development of a wearable assistive device for navigation for the visually impaired with command and request support* (PhD Thesis). Soka University, California.

[14] Bhatlawande, S., Borse, R., Solanki, A., & Shilaskar, S. (2024). A smart clothing approach for augmenting mobility of visually impaired people. *IEEE Access, 99*, 1–11.

[15] Masadeh, M., Alkhdour, E., AbuDiak, H., & Obaidat, R. (2024). Approximate computing-based assistive shopping trolley for visually challenged people. *International Journal of Computing and Digital Systems, 15*, 1405–1416. https://doi.org/10.12785/ijcds.2024.15.1405.

[16] Hoogsteen, K. M., Szpiro, S., Kreiman, G., & Peli, E. (2022). Beyond the cane: Describing urban scenes to blind people for mobility tasks. *ACM Transactions on Accessible Computing (TACCESS), 15*, 1–29.

[17] Kostyuchenko, N., & Smolennikov, D. (2022). Social responsibility for sustainable development goals. In *Reducing Inequalities Towards Sustainable Development Goals*. River Publishers.

[18] Oerther, S. E., & Rosa, W. E. (2020). Advocating for equality: The backbone of the sustainable development goals. *AJN The American Journal of Nursing, 120*, 60–62.

[19] United Nations. (2024, August 20). *The 17 goals*. https://sdgs.un.org/goals.

[20] Hoogsteen, K. M., Szpiro, S., Kreiman, G., & Peli, E. (2022). Beyond the cane: Describing urban scenes to blind people for mobility tasks. *ACM Transactions on Accessible Computing (TACCESS), 15*, 1–29.

[21] Mai, C., Xie, D., Zeng, L., Li, Z., Li, Z., Qiao, Z., & Li, L. (2022). Laser sensing and vision sensing smart blind cane: A review. *Sensors, 23*(2), 869. https://doi.org/10.3390/s23020869.