

AN INTELLIGENT SEQUENTIAL NAVIGATION FRAMEWORK FOR OBSTACLE PROGRESSION ANALYSIS AND RISK-AWARE ASSISTIVE GUIDANCE FOR VISUALLY IMPAIRED INDIVIDUALS

D. KARTHIKA¹, *S.P. BALAMURUGAN²

¹Research Scholar, Department of Computer and Information Science,
Annamalai University, Tamil Nadu, India

²Assistant Professor/Programmer, Department of Computer and Information Science,
Annamalai University, Tamil Nadu, India

E-mail: ¹karjayamca19@gmail.com, ²spbcdm@gmail.com

ABSTRACT

Traditional assistive navigation systems for visually impaired people evolved early pieces of research on the understanding of sequential obstacles, adaptation of movements to adjust risk levels and reasoning about intelligent navigation. To overcome these limitations, this paper present an intelligent assistive navigation framework called SNAP-RAG (Sequential Navigation and Progression-based Risk-Aware Guidance) that integrates lightweight obstacle perception, sequential obstacle progression analysis, adaptive risk scoring, context-aware navigation decision generation and voice-guided assistance into a single real-time framework. First, in light of the computationally efficient obstacle detection under various navigation environments, a Lightweight Spatial Attention Feature Boosting (LSAFB) model based on YOLOv5n is designed. Then, Sequential Obstacle Progression Module (SOPM) examines spatial obstacle progression in order to estimate relative distance and directional position. A novel Adaptive Navigation Risk Scoring (ANRS) framework is then proposed to assess obstacle severity granularity-wise using spatial, positional, and progression-aware contextual features. Also, a Context-Aware Navigation Decision Engine (CNDE) combines obstacle risk and environmental context before providing intelligent movement guidance in order to make safe navigation decisions. Lastly, an assistive voice guidance module executes the generated navigation directions as immediate speech responses to engage with users. The experimental results validate that the proposed framework realizes prominent obstacle understanding, navigation robustness as well as assistive guidance performance against state-of-the-art models and with a lightweight computational complexity for CPU-based real-time deployment. The proposed framework provides an efficient and scalable approach for intelligent assistive navigation in the visually impaired environment.

Keywords: *Assistive Navigation System, Sequential Obstacle Progression Risk-Aware Navigation, Obstacle Detection, Context-Aware Guidance, Visually Impaired Assistance, Intelligent Mobility Framework, Voice-Guided Navigation*

1. INTRODUCTION

Visual impairment has a significant impact on an individual's independence and mobility during navigation of complex indoor and outdoor environments. Traditional technologies, including white canes and guide dogs have very limited perception of the surroundings and cannot detect dynamic obstacles, spatial hazards or contextual navigation difficulties in real time. Therefore, the intelligent assistive navigation systems which combine computer vision, and deep learning in addition to online guidance mechanisms have been one of hot active research topics over recent years [1]. Currently, assistive systems are limited to obstacle detection, scene understanding and

object localization in order to make navigation safe for visually impaired users.

Recently, deep learning-based object detection frameworks have shown promising uplift in environment perception tasks for assistive navigation applications. Said et al. SomeWorks based on [1] proposing a deep learning framework using neural architecture search, excels in both detection accuracy and computational efficiency for visually impaired navigation assistance. Similarly, Atitallah et al. A lightweight embedded deployment enabled modified YOLOv5 based indoor and outdoor obstacle detection framework [2]. In addition, compressed YOLO based frameworks for detecting obstacles during real-

time perception tasks in assistive navigation architectures form reasonably well [3] further confirming the hopes of wearable based navigation assistance systems continuing to exhibit this level of performance. These methods illustrate that lightweight deep learning models are applicable to obstacle perception in practical assistive scenarios. Furthermore, some recent studies have also investigated intelligent navigation assistance mechanisms for the blind. Kashif et al. [4] developed a smart obstacle avoidance framework involving object detection and navigation guidance in advance for the autonomy to travel safely through dynamic environments. Similarly, all image captioning and scene description approaches have also been used to estimate semantic environment understanding for outdoor navigation assistance [5]. More recently, vibrotactile feedback-based navigation assistance systems towards directional obstacle perception have been proposed, such as GuideTouch [6]. There are also mixed-reality-based navigation assistance frameworks to guide visually impaired users in a way that increases contextuality of environmental interaction [7].

However, current assistive navigation systems still have many limitations. Existing approaches mostly attend to obstacle detection and basic localization, while neglecting sequential progression towards obstacles, contextual navigation reasoning and adaptive risk estimation. In real-world navigation scenarios, just recognizing an obstacle will not be enough to produce safe and the context of assistive guidance. Instead, it is very crucial for the consistent operation of navigation assistance to recognize whether an obstacle situated in front of, behind, directly below or above the ego vehicle could further endanger following traffic. Additionally, most of the existing systems are based on computationally expensive architectures, task-specific hardware sensors or GPU-based heavy frameworks which reduces their applicability for lightweight real-time deployment in low-resource environments [8]. A related challenge is the lack of integrated navigation intelligence in any assistive paradigms to date. Most existing systems process single frames independently, neglecting the temporal progression of obstacles and complex interactions with contextual components in their environment. Thus these systems do not offer adaptive navigation decisions that take into account obstacle severity, direction of travel, and dynamic

spatial relationships. Furthermore, research on assistive navigation of visually impaired people has largely under explored risk-aware navigation assistance and intelligent movement recommendation mechanisms.

To overcome these limitations, this paper proposes an intelligent assistive navigation framework named SNAP-RAG. This framework includes a lightweight obstacle detection component for localized navigation information collection, sequential analysis of identified obstacles and their progression times, an adaptive scoring of navigation risk based on continuous risk monitorization, context-aware structured assembly of choices and decisions to provide a real-time voice assistant with various levels of output from this agent. First, a computationally effective obstacle detection method with diverse navigation environments based on the technique of LSAFB framework for YOLOv5n is built. Then, a SOPM is proposed to predict the distance and direction the blind progress toward in obstacles by ordering obstacle sequences. Moreover, this work proposes a novel ANRS framework that evaluates obstacle risk levels based on various contextual features - spatial, positional and progression-aware. Finally, the CNDE creates smart movement during navigation direction generation and speech-based navigation support for safe and adaptive navigation assistance. Experimental results show how the proposed framework significantly outperforms current state-of-the-art methods in terms of obstacle perception, navigation robustness, contextual risk assessment and assistive guidance performance while being computationally feasible for real-time applications.

Research Objectives and Research Questions

Based on the identified research gaps, this study is guided by the following objectives:

O1: To develop a lightweight obstacle perception mechanism capable of accurately detecting and localizing navigation-relevant obstacles under CPU-constrained conditions.

O2: To develop a sequential obstacle progression mechanism for estimating relative obstacle distance and directional position from ordered navigation frames.

O3: To develop an adaptive navigation risk-scoring mechanism that incorporates obstacle

category, spatial position, proximity, and progression-related contextual information.

O4: To develop a context-aware navigation decision mechanism that converts perception and risk information into actionable movement recommendations and real-time voice guidance.

O5: To evaluate whether the complete framework can provide accurate, computationally efficient and real-time assistive navigation without requiring GPU-based deployment.

These objectives lead to the following research questions:

RQ1: Can lightweight spatial-attention-based obstacle perception provide sufficiently accurate obstacle recognition for assistive navigation under CPU-constrained deployment?

RQ2: Can sequential analysis of ordered obstacle observations provide useful relative distance and directional progression information beyond single-frame obstacle detection?

RQ3: Can adaptive integration of obstacle severity, spatial position, proximity and progression improve contextual navigation-risk assessment?

RQ4: Can the integration of obstacle perception, progression analysis and risk assessment generate context-aware movement recommendations and real-time voice guidance?

RQ5: Can the complete SNAP-RAG framework achieve this integrated functionality while maintaining computational efficiency suitable for CPU-based real-time deployment?

2. RELATED WORKS

The technological advancements in computer vision and deep learning have made huge strides in intelligent assistive navigation systems for blind people. Current studies have focused on obstacle detection, wearable navigation aids, spatial perception, risk assessment and real time environmental perception. Yet combining sequential reasoning of obstacle progression analysis, adaptive navigation inference and context-based guidance generation is still relatively unexplored.

Deep learning-based obstacle detection frameworks have been popular in assisting visually-impaired navigation since they perform

robust real-time environment perception. Wang et al. [9] introduced a lightweight deep learning-based model called YOLO-OD framework for obstacle detection- based navigation assists, showing better performance of recognizing obstacles in practical navigation cases. For instance, Obayya et al. [10] presented an intelligent indoor assistive system based on the combination of YOLO and recurrent neural networks to understand obstacles and provide navigation assistance. Although they improved the obstacle perception capabilities of their framework, their system was primarily based on obstacle recognition for indoor objects without responsive navigation intelligence. The recent years have also seen a lot of interest in wearable assistive navigation systems. On the other hand, Shah et al. [11] proposed a vision-based smart wearable navigation framework using deep learning approaches for obstacle recognition and direction guidance. The system was meant to be worn as a device by visually impaired users and worked successfully in respect to obstacle awareness but failed to acknowledge contextual risk based reasoning and sequential progression between obstacles. In 2026, Mahapatro et al. [12] proposed a real-time open vocabulary spatial navigation system utilizing YOLO-World and monocular depth estimation for localizing obstacles. This framework demonstrated strong navigation performance, however the system includes computationally expensive depth estimation methods that may be impractical for lightweight deployment. There are also some studies targeting intelligent navigation support using monocular depth estimation and spatial awareness mechanisms. The smart assistive navigation system for visually challenged people which follows object detection along with monocular depth estimation was proposed by Okolo et al. [13]. In their paper, Nagarajan and Gopinath [14] proposed a hybrid optimization-enabled deep learning framework for object classification of indoor objects along with estimating the distance. Their framework enhanced the distance prediction accuracy via optimization strategies, but did not produce contextual navigation guidance and risk-aware reasoning. Smartphone-based assistive navigation systems are another effective method for practical implementation. DeepNAVI - Kuriakose et al. [15] developed a smartphone-based navigation assistance for blind people using deep learning to understand obstacles and assistive-oriented navigation. Likewise, Joshi et al. [16] developed

low-cost real-time assistive framework for visually impaired people which can help in sensing and interpret the environment using AI-SenseVision. The solution was intended with respect to light deployment and computational efficiency but did not innovate upon adaptive obstacle severity analysis or intelligent movement recommendation mechanisms.

Recent studies examined advanced YOLO-based frameworks of obstacle perception for indoor and outdoor adaptive navigations. Indoor-outdoor obstacle detection method proposed for YOLO In-Out with adaptive spatial pooling squeeze and attention YOLO framework by Gladis et al. [17]. Their model enhanced the representation of spatial features and also increased robustness for obstacle detection. Xiang and Sanie [18] created a spatial awareness system to facilitate indoor navigation and daily living activities through computer vision-based environmental understanding. While their framework enhanced spatial perception, it was mainly aimed at indoor environmental assistance. In addition, reinforcement learning and intelligent navigation decision-making strategies have also grown research attention. The sidewalk obstacle avoidance framework with deep reinforcement learning by Ahmed et al. [19]. The framework showed adaptive movements learning in dynamic environments, but sequential obstacle progression interpretation and spatial reasoning with contextual navigation were not incorporated. Bamdad et al. [20] proposed a vision-based SLAM navigation system which also provided environment mapping and localization support. The framework did improve navigation awareness, but SLAM-based architectures remain computationally expensive to run in practice for small footprints. In fact, recent review studies have summarised those recent advances in assistive navigation technologies. Yao et al. [18] focused on vision-based assistive systems for the blind and mentioned that lightweight deep learning, wearable systems and contextual environmental understanding approaches have become more important. Similarly, Ikram et al. [22] also proposed an IoT-based obstacle detection and warning system based on sensor-based environmental perception for assistive mobility. According to Lin et al. [23], they also proposed a wearable, deep learning-based assistive framework for fast obstacle perception and navigation assistance, thus prove the

possibility of using AI based wearable to assist navigation.

2.1 Critical Synthesis of Recent Assistive Navigation Research

Assistive navigation research is evolving away from traditional obstacle identification and towards spatial/deployment awareness. Studies are being conducted to improve the ability to extract features from images, identify small objects within images, and increase computational efficiency. Examples of such studies include: PC-CS-YOLO, which reduces computational overhead for real-time obstacle avoidance [24]; and YOLO-Extreme, which provides increased robustness of perception in foggy environments [25]. While advances have been made in improving the accuracy of perception of obstacles visually, most of the research conducted is aimed at enhancing the perception itself, as opposed to developing sequential, risk-based decision-making models for navigating through those perceptions. Recent systems emphasize the importance of explicit distance perception in assistive navigation. For example, Shah et al. designed an assistive navigation system that incorporates object detection, distance estimation, and voice feedback into a single wearable embedded platform [11]. Similarly, SGBM_YOLO is a method that uses stereo vision with YOLO to estimate the depth of objects, which can achieve less than 3.2% measurement error [26]. However, using depth sensors or other methods that require additional hardware or computation to determine the distance to an object, does not provide adaptive risk interpretation or context-based movement reasoning. Multimodal navigation systems, such as MR.NAVI, have integrated several types of sensors, evaluation tools, and feedback mechanisms [7]. Although this represents a movement toward multimodal, user-centered navigation, it also creates additional complexity. Therefore, there is still a need for lightweight frameworks that can support the progression of obstacles, risk assessment, and provide assistive feedback to users without requiring special hardware.

Despite tremendous advancements in obstacle detection and assistive navigation revealed through solving existing studies, there are many critical research gaps not previously covered. Existing methods mostly concern static obstacle

perception and elementary distance estimation while neglecting sequential trajectory analysis, adaptive navigation risk scoring and context-aware navigation reasoning. Additionally, a number of existing frameworks are based on computationally intensive architectures, or require the use of specialized hardware devices or sensors making lightweight real-time deployment impractical in real world applications.

2.2 Explicit Research Gaps and Motivation for the Present Study

Recent studies have shown that there are many other unresolved issues with regard to both the perception-to-navigation decision making process as well as the ability to accurately detect obstacles; these include the resolution of four specific "grey areas".

Gap 1 – Lack of sequential obstacle progression reasoning:

Most recent vision systems have focused on identifying obstacles within the current image or video frame. Although some depth-aware approaches provide better spatial information, most of them do not track obstacle movements in relation to the user's intended route. Consequently, there is still a significant disconnect between what users can see at any given time (perception) and how they can use that visual data to guide themselves along their path (navigation).

Gap 2 – Limited adaptive risk interpretation:

Risk assessment in current systems uses fixed criteria (thresholds), however, assessing risk when navigating for the visually impaired needs to be based on context, such as what type of obstacle it is; how close it is to the user; and where the user is heading. Although sensors have advanced, interpreting the risk based on the situation has not been integrated into the navigation system for the visually impaired.

Gap 3 – Insufficient integration of perception, risk and movement decision-making:

While independent developments have been made in object detection, depth estimation, wearable feedback, and scene understanding; there is no current development of an integrated lightweight

pipeline to convert the observed obstacles into motion control commands.

Gap 4 – Trade-off between navigation intelligence and computational simplicity:

Although contemporary advanced performance navigation systems enhance capabilities, many of these systems will necessitate supplemental sensor and computing capacity. As such, there continues to be an opportunity to develop a singular system that provides sequential reasoning, adaptive risk evaluation, and context dependent direction while operating within the limits of low resource CPU architectures.

Research need derived from these gaps:

The SNAP-RAG system takes a different approach to address previous limitations in object detection systems. The system includes four main modules as part of its navigational assistance pipeline: The Sequential Obstacle Progression Module (SOPM), which performs sequential analyses; The Adaptive Navigation Risk Scoring (ANRS), which evaluates risks; The Context-Aware Navigation Decision Engine (CNDE), which provides contextual information; And a lightweight, CPU-based LSAFB-YOLOv5n architecture, used to perform the perception tasks.

To address these limitations, this work proposes the SNAP-RAG framework: a novel light-weight intelligent assistive navigation architecture which involves sequential obstacle progression analysis, adaptive navigation risk scoring, context-aware navigation decision generation and voice guided assistance within one unified real-time frame of reference to design. In contrast to the state of the art approaches, the proposed framework is capable of realizing obstacle progress awareness, contextual navigation knowledge and computationally efficient deployment for physically realizable guidance systems for visually impaired individuals.

3. THE PROPOSED MODEL

In this proposed SNAP-RAG framework, a lightweight multi-stage assistive navigation method is developed to combine obstacle detection with progressive navigation through obstacles based on risk assessment and system context awareness through generated decisions and voice assistance to visually impaired users.

First, a computationally-efficient obstacle detection and localization are performed from the navigation scenes through an LSAFB network based on YOLOv5n. Meanwhile, the SOPM processes ordered obstacle sequences to measure the mature distance of obstacles and their locations in terms of directions. Additionally, the ANRS framework assesses difficulty of obstacles through contextual features about position, space

and progress-awareness. Next, the CNDE processes the generated contextual information to generate intelligent motion guidance and adaptive navigation decisions. An assistive voice guidance module then converts these navigation instructions into speech assistance to assist the user as a safe mobility assistant in real-time. Figure 1 provides the block diagram of the proposed SNAP-RAG framework.

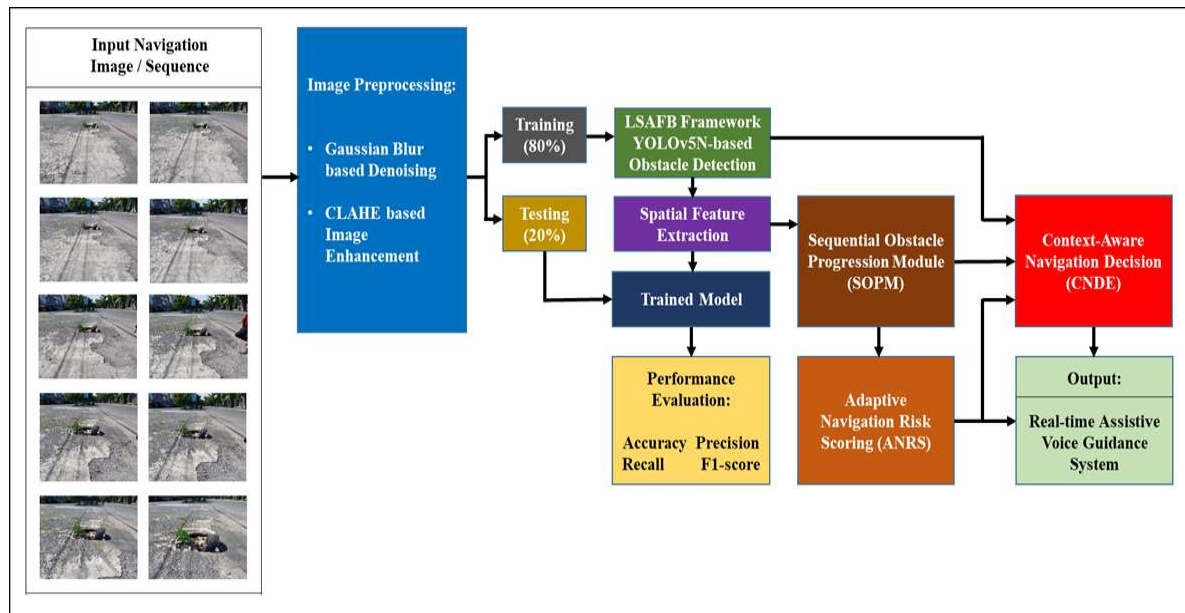


Figure 1: The Block Diagram of the proposed SNAP-RAG Framework

3.1 Dataset Collection and Organization

This work used a dataset for obstacle navigation collected specifically to evaluate several tasks involving the detection and progression of obstacles, predicting risks, and guiding navigation without vision. It consists of real indoor, and outdoor navigation videos taken in a variety of environments. It included 22 obstacles (right turn, left turn, puddle, street vendor, obstacle, bad road, garbage bin, chair, pothole car motorcycle pedestrian fence gate barrier roadblock door tree plant pot drain stair pole zebra cross). For each image, there is an annotated file in YOLO format with a (class, center x, center y, width and height) values. Initially, the dataset was prepared with training, validation and testing subsets which consist of separate image and label folders for proper modeling approach by systematic model development pipeline.

An additional sequence-aware data structure organization strategy was additionally proposed, to assist the sequential live obstacle progression analysis task. Obstacle images were organized into sequence folders according to continuity and relative advancement. Then, the sequences were reversed to replicate real-time navigation motion from far away obstacle locations to close up obstacle space. The initial dataset was utilized to train an obstacle detection model, and the generated sequential dataset was used for position estimation, risk prediction, and navigation guidance generation through SNAP-RAG. Table 1 shows an overview of the dataset used, and a representative sample of the different classes is shown in Figure 2.

Table 1: Description of Dataset

Parameter	Description
Dataset Source	Kaggle
Total Images	3355
Number of Classes	22
Annotation Format	YOLO
Dataset Structure	Train / Validation / Test
Training Images	70%
Validation Images	15%
Testing Images	15%
Sequence Dataset	Yes
Sequence Organization	Ordered Obstacle Progression
Input Image Format	JPG
Label Format	TXT



Figure 2: Sample images of different classes

3.2 Sequential Dataset Generation

A framework is proposed in this work upon which a sequential dataset can be generated in a way that enables obstacle progression analysis and navigation-aware contextual understanding. First, the obstacle images and its related YOLO annotation files were gathered from the training, validation and testing subset, organized in a serial order. Then these ordered images were placed into specific sequence folders for groups of continuous obstacle progression scenarios.

In addition, a reverse progression arrangement strategy was presented to emulate constant backstepping motion from remote impedances toward close burdens. In this setting, frames that contain distant obstacles will show up at the start of the sequence followed by those containing nearby obstructions at a later time. This sequential dataset was used in SOPM for calculating the distance to obstacles, direction of movement and contextual navigation-aware understanding. A sample obstacle progression sequence representing ordered navigation frames from far to near obstacle positions is shown in Figure 3.



Figure 3: Single Obstacle Progression Sequence Showing Ordered Navigation Frames from Far to Near Obstacle Positions

3.3 Image Preprocessing and Enhancement

Image pre-processing and enhancement was done to make the obstacles stand out better and for noise colors interference to reduce, by applying contrast adjustment, colour filtering in HSI space, Gaussian blurring, as these processes affect shape features. Initially, a new input size of 416×416 pixels was chosen for computational efficient model training and feature extractor putting all images to the same spatial resolution.

Then, denoising with Gaussian blur was adopted to remove the unwanted noise and minor intensity fluctuations in the navigation images. The Gaussian filtering operation is mathematically represented as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

Where $G(x, y)$ denotes the Gaussian kernel function and σ represents the standard deviation controlling the smoothing intensity. The Gaussian filter enhances structural coherence, while maintaining the boundaries of significant obstacles.

In addition, Contrast Limited Adaptive Histogram Equalization (CLAHE) was implemented to maximize the local contrast level of obstacles in different illumination environments. The contrast enhancement operation can be expressed as:

$$I_{enh}(x, y) = CLAHE(I(x, y)) \quad (2)$$

Where $I(x, y)$ represents the input image and $I_{enh}(x, y)$ denotes the enhanced output image after adaptive histogram equalization. CLAHE works due to its localized nature, causes major improvement in texture detection. However, excessive contrast amplification is avoided.

Finally, pixel intensity normalization was done to scale all of the pixel inputs into $[0,1]$, which enables a stable and efficient learning for any converging model as well. The normalization process is given by the following equation:

$$I_{norm} = \frac{I}{255} \quad (3)$$

Where I denotes the input pixel intensity and I_{norm} represents the normalized image intensity. The sample original images and its pre-processed versions are provided in Figure 4.



Figure 4: Sample Original and Preprocessed Images after Gaussian Blur Denoising and CLAHE-Based Contrast Enhancement

The preprocessed images were then sent to the proposed LSAFB framework for obstacle classification and spatial feature extraction.

3.4 LSAFB-Based Lightweight Obstacle Detection Framework

In this work, a novel Lightweight Spatial Attention Feature Boosting (LSAFB) framework was developed to accomplish obstacle detection in a computational cost-effective manner for visually impaired navigation assistance. The proposed framework leverages the YOLOv5n architecture along with spatial-attention based feature refinement to enhance the representation of obstacles and support contextual multi-scale degradation. The framework runs obstacle localization and classification along with

minimum computation that it can be deployed on CPU based deployment environments in real-time.

3.4.1 YOLOv5n backbone

Here, YOLOv5n backbone network was chosen as the main structure design of obstacle detection because it is easy to deploy, has fewer parameters and can infer in real time. The YOLOv5n architecture has fewer network parameters and much lower memory consumption than conventional deep object detection architectures, resulting in a framework that is ideally suited for deployment of assistive navigation systems under computationally constrained environments.

To begin with the preprocessed navigation image is passed down to the backbone network for hierarchical feature extraction. The backbone carries out a series of layered operations to learn obstacle specific spatial information through convolutions, feature aggregation, and multi-scale representation learning. The convolution operation is mathematically represented as:

$$F(i, j) = \sum_m \sum_n I(i - m, j - n) \cdot K(m, n) \quad (4)$$

Where I denotes the input feature map, K represents the convolution kernel, and $F(i, j)$ indicates the extracted feature representation at spatial location (i, j) .

The extracted features are then passed through the neck and detection head to localize and classify obstacles. Lightweight architecture of YOLOv5n backbone makes it possible to obtain efficient obstacle detection results with a lower inference latency suitable for CPU-based real-time navigation assistance.

3.4.2 Spatial attention feature boosting

A Spatial Attention Feature Boosting strategy is integrated inside the lightweight detection framework, which helps enhance ability for obstacle representation. The proposed attention mechanism can accentuate spatial regions which are important to the obstacle of interest and down-weight background distraction as well as other environmental information, irrelevant in this context.

The extracted feature maps are firstly fed with average pooling and maximum pooling operations as they both capture complementary spatial

information. The pooled features are then concatenated and propagate to a convolution layer to obtain the spatial attention map. The spatial attention operation is expressed as:

$$M_s(F) = \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)])) \quad (5)$$

Where $M_s(F)$ denotes the generated spatial attention map, σ represents the sigmoid activation function, and $f^{7 \times 7}$ indicates the convolution operation applied over the concatenated pooled features.

Specifically, the produced attention map is then multiplied with the initial feature representation to put particular importance on obstacle-specific regions for enhanced features refinement. Moreover, in the proposed method, at first, multi-scale obstacle representation learning is realized to achieve effective detection of small, medium and large navigation obstacles under the changeable environmental conditions.

3.4.3 Bounding box prediction

The refined feature maps are then passed on to the YOLO detection head, where each obstacle is localized and a confidence score as well as class prediction is estimated. This detection head predicts the coordinates of the center of each obstacle, its bounding box dimensions and a confidence probability for every detected obstacle. The bounding box prediction process is represented as:

$$B = (x, y, w, h, c) \quad (6)$$

Where (x, y) denote the center coordinates of the obstacle, w and h represent the bounding box width and height, and c indicates the confidence score associated with obstacle detection.

The object confidence score is calculated using the sigmoid activation function:

$$C = \frac{1}{1 + e^{-x}} \quad (7)$$

Where C denotes the predicted confidence probability for obstacle existence.

Then, the classification probability for each obstacle category is assessed using the softmax function:

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^N e^{z_j}} \quad (8)$$

Where $P(y_i)$ denotes the predicted probability of the i^{th} obstacle class and N represents the total number of obstacle categories.

For the end-to-end training of detection framework for better obstacle localization and classification performance, we use a composite loss function with the integration of localization loss confidence loss and classification loss overall:

$$L_{total} = L_{box} + L_{conf} + L_{cls} \quad (9)$$

Where L_{box} , L_{conf} , and L_{cls} denote the bounding box regression loss, confidence loss, and classification loss, respectively.

The proposed LSAFB framework can produce (i) accurate obstacle localization results and (ii) multi-class obstacle predictions that are fundamental to the subsequent steps of obstacle progression analysis, adaptive risk estimation, and intelligent navigation guidance generation within the framework of the proposed SNAP-RAG. The sample obstacle detection results generated by the proposed LSAFB framework are presented in Figure 5.



Figure 5: Sample Obstacle Detection Results Generated by the Proposed LSAFB Framework

3.5 Sequential Obstacle Progression Module (SOPM)

In this work, a SOPM was built to investigate obstacle progression behavior by estimating the distance and directional position of an obstacle for intelligent assistive navigation. Unlike traditional obstacle detection frameworks that analyze individual images, the proposed SOPM uses ordered navigation frames to achieve sequence-aware obstacle understanding.

In the first step, previously detected obstacle sequences were assessed based on their temporal continuity over successive frames. The bounding box variation and relative spatial movement are implemented to do obstacle progression analysis,

because obstacle position is determined during navigation movement by this way.

The obstacle area was computed using $A = w \times h$, where w and h denote the width and height of the detected obstacle bounding box, respectively. The obstacle progression behavior was estimated based on the relative variation in obstacle area across consecutive frames:

$$P = \frac{A_t - A_{t-1}}{A_{t-1}} \quad (10)$$

Where A_t and A_{t-1} represent the obstacle areas in the current and previous frames, respectively, and P denotes the progression ratio. An increasing progression ratio indicates obstacle proximity movement toward the user.

Moreover, an estimation of obstacle distance was made through spatial obstacle area variation. The model was trained on progression features, computed from data, to recognize the obstacles across three navigation-aware distance classes:

$$D = \begin{cases} \text{Far}, & A < T_1 \\ \text{Medium}, & T_1 \leq A < T_2 \\ \text{Near}, & A \geq T_2 \end{cases} \quad (11)$$

Where A denotes the obstacle area and T_1 and T_2 represent predefined distance thresholds.

To estimate obstacle direction, the horizontal center position of the detected obstacle was computed using:

$$C_x = \frac{x+w/2}{w} \quad (12)$$

Where x denotes the bounding box horizontal coordinate, w represents the bounding box width,

and W indicates the image width. Based on the normalized horizontal center position, the obstacle direction was estimated as:

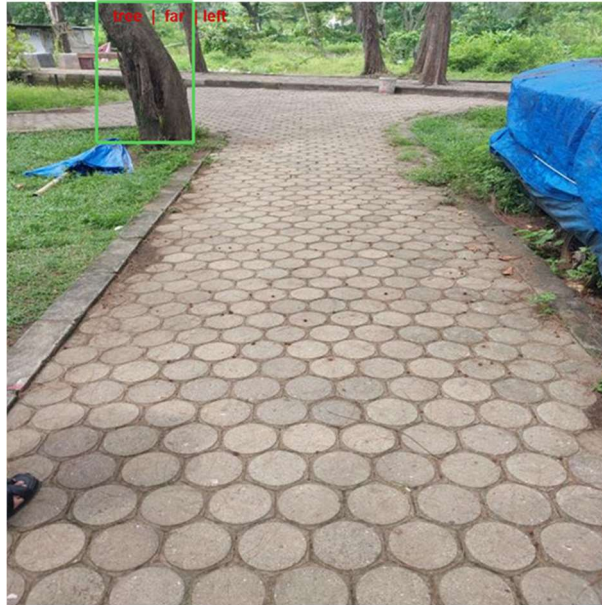
$$Dir = \begin{cases} \text{Left}, & C_x < 0.33 \\ \text{Center}, & 0.33 \leq C_x < 0.66 \\ \text{Right}, & C_x \geq 0.66 \end{cases} \quad (13)$$

Sample navigation images corresponding to different example obstacles distance and directness conditions used in the SOPM framework proposed in this paper are shown in Figure 6. In particular, the first image indicates a far-left obstacle location, second image illustrates a medium-center obstacle location and third image depicts a near-right obstacle point. Different spatial variations are used for obstacle progression analyzes, distance calculation, and navigation-aware directional reasoning.

Obstacle: tree

Distance: far

Direction: left



Obstacle: pothole

Distance: medium

Direction: center



Obstacle: garbage bin

Distance: near

Direction: right



Figure 6: Sample Obstacle Progression Images Representing Different Obstacle Distances and Directional Positions

The proposed SOPM combines obstacle progression behavior, distance estimation and directional reasoning to enable analysis of sequence-aware navigation behaviour, wherewith the output is utilized by SNAP-RAG to offer adaptive navigation guidance.

3.6 Adaptive Navigation Risk Scoring (ANRS)

In this work, ANRS framework was created to estimate obstacle severity and determine navigation danger levels for the visually impaired assistance. In this paper, a framework is proposed to do contextual danger analysis through information from obstacle category, spatial position, and obstacle movement behaviors with distance-estimation features.

Initially, the feature sets extracted based on the SOPM that are relevant to navigation including obstacle class, obstacle area, direction position, and distance estimate level. Adaptive weights for obstacle severity were computed based on these contextual features and specific criteria based on the total navigation danger that an individual category of obstacle at a particular distance may signify. Contextually, any obstacles located at the center of the way were also given more weight in contextual danger importance. Subsequently, the overall navigation risk score was computed by integrating obstacle severity, spatial position influence, and contextual progression importance using:

$$R = W_o + W_s + W_p \quad (14)$$

Where R denotes the final navigation risk score, W_o represents the obstacle severity weight, W_s indicates the spatial position weight, and W_p denotes the progression-aware contextual weight obtained from the SOPM framework.

Based on the computed risk score, the obstacles were categorized into four navigation-aware danger levels:

$$Risk = \begin{cases} Low, & R < T_1 \\ Medium, & T_1 \leq R < T_2 \\ High, & T_2 \leq R < T_3 \\ Very High, & R \geq T_3 \end{cases} \quad (15)$$

Where T_1 , T_2 , and T_3 denote predefined risk thresholds.

Figure 7 illustrates the navigation risk categorization levels utilized in the proposed ANRS framework.





Figure 7: Risk Categorization Levels of the Proposed ANRS Framework

The proposed ANRS framework simultaneously performs the adaptive contextual danger analysis based on obstacle proximity, obstacle severity, and spatial movement behavior. The risk information generated is then utilized by the CNDE to generate intelligent navigation guidance in the proposed SNAP-RAG framework.

3.7 Context-Aware Navigation Decision Engine (CNDE)

This work developed a CNDE to produce intelligent movement guidance and safe navigation directions for people with visual impairments. In this paper, the proposed framework conducts contextual navigation reasoning by combining the obstacle distance and directional position information with adaptive risk

that was extracted from the SOPM and ANRS frameworks.

The data were first analysed based on the contextual features of the obstacle which include distance level, directional position and navigation risk category to calculate how most safely to navigate. When generating the decision for where to move, obstacles located nearby and categorized as having a high risk level were given higher priority of movement. The contextual navigation score was computed using:

$$N = W_d + W_r + W_p \quad (16)$$

Where N denotes the navigation decision score, W_d represents the directional movement weight, W_r indicates the adaptive risk weight, and W_p denotes the obstacle proximity weight.

CNDE framework is to generate navigation commands in either move left, move right, move forward, slow down, stop and safe-forward follow the recommendation with regard to the calculated navigation score and obstacle registration. The navigation decision logic is represented as:

$$Nav = \begin{cases} \text{Move Right,} & \text{Obstacle = Left} \\ \text{Move Left,} & \text{Obstacle = Right} \\ \text{Slow Down,} & \text{Risk = High} \\ \text{Stop,} & \text{Risk = Very High} \\ \text{Move Forward,} & \text{Risk = Low} \end{cases} \quad (17)$$

The context navigation decision logic used in the CNDE framework is depicted in Figure 8, where obstacle position and adaptive risk information are used to build safe movement (move left, move right, slow down and stop) guidance.

Obstacle: chair Distance: near Direction: left Risk: high Navigation Decision: move right	
Obstacle: garbage bin Distance: medium Direction: right Risk: high Navigation Decision: move left	
Obstacle: pothole Distance: near Direction: center Risk: high Navigation Decision: slow down	

Obstacle: obstacle Distance: near Direction: center Risk: very high Navigation Decision: stop	
Obstacle: pothole Distance: far Direction: center Risk: low Navigation Decision: move forward	

Figure 8: Navigation Decision Logic of the Proposed CNDE Framework

A contextual navigation intelligence provides the following by combining obstacle progression behavior, and spatial positioning with adaptive risk analysis for real time assistive navigation path recommendations. The calculated navigation decisions are subsequently sent to the assistive voice guidance module that provides real time audio based navigation assistance.

3.8 Assistive Voice Guidance System

This work presents the creation of a real-time audio-based navigation assistance for visually impaired people called Assistive Voice Guidance System. The framework converts contextual navigation information extracted by the SOPM, ANRS and CNDE frameworks into useful verbal cues for mobility assistance.

Initially, contextual obstacle information such as obstacle classification, distance level I, position, navigation risk and the advising movement was combined to create assistive navigation text. The generated navigation instruction is represented as:

$$T = \{O, D, Dir, R, Nav\} \quad (18)$$

Where T denotes the generated assistive text instruction, O represents the obstacle category, D indicates the estimated obstacle distance, Dir denotes the obstacle direction, R represents the predicted navigation risk, and Nav indicates the generated navigation decision.

To do this, a mechanism of text-to-speech synthesis was used to generate speech signals from the output assistive text generated. The

speech synthesis process is represented as, $S = f(T)$, where S denotes the generated speech output and $f(T)$ represents the speech synthesis function applied over the generated navigation text instruction.

The proposed framework enables assistive narration through real-time navigation-aware dynamic generation of instructions for safe movement, including move left, move right, slow down, stop, and the direction to move forward safely. Moreover, voice interaction support allows for uninterrupted navigation recommendations, by sequentially constructing the environmental understanding and generating adaptive contextual guidance.

The synthesized assistive speech guidance is required to be concise to support real-time deployment within SNAP-RAG, and offers a noticeable improvement in awareness of both navigation support and environment understanding for a visually impaired user.

3.9 GUI-Based Deployment Framework

In this work, a GUI-Based Deployment Framework was developed to give an interactive assistive navigation interface in real-time for visually impaired users. The developed graphical user interface (GUI) encompasses all components under a common deployment umbrella, including obstacle detection, obstacle progression analysis, risk prediction, generation of navigation guidance and voice advice.

For systematic visualization of outputs produced during the iterative stages of the proposed SNAP-RAG framework, a 2×3 grid-based interface structure was used to best design the GUI. The first row of the interface contains original navigation image, obstacle detection results and

obstacle position estimation outputs, while results from risk prediction, navigation decision outputs and assistive voice guidance information is indicated in the second row of the interface.

In addition, a menu-driven navigation format was added to facilitate the execution of frameworks in sequential order. Options in GUI include different fields like Load Image, Detect Obstacle, Detect Position, Predict Risk Navigate Decision, Voice Assistant and Exit. It activates one module at a time, based on the execution flow of the proposed framework, ensuring structured navigation analysis and controlled real-time deployment.

Furthermore, the proposed deployment framework includes a lightweight CPU-compatible executable training that can be used for real-time understanding of obstacles in Robot application and generation of contextual navigation guidance. The generated assistive guidance is displayed within the interface and converted into voice narration output interaction-based navigation assistance. By providing a modular and interactive design, while also easily visualizing the data in real-time, the presented GUI framework increases usability and interpretability of navigation applications for visually impaired individuals while assisting practical implementation.

The 2×3 GUI developed for real-time deployment of the proposed SNAP-RAG framework are presented in Figure 9, where outputs such as obstacle detection, position estimation, risk prediction, navigation decision generation, and assistive voice guidance are visualized interactively.

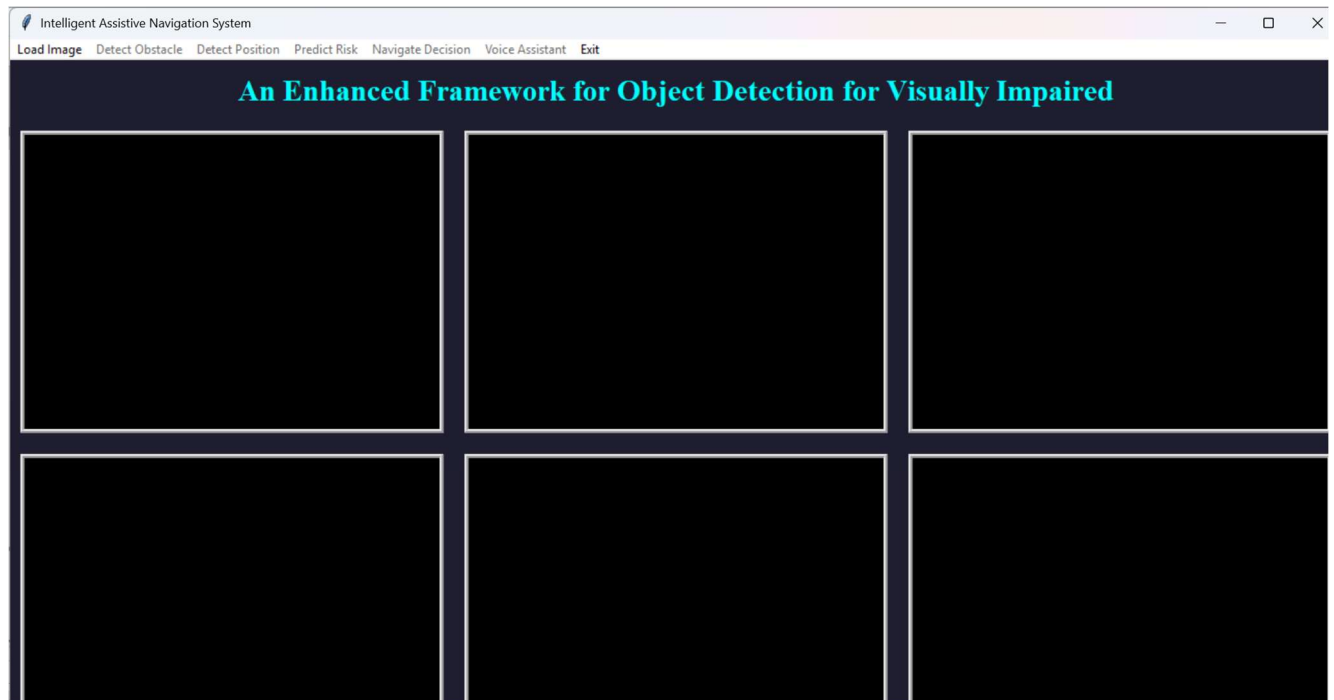


Figure 9: GUI-Based Real-Time Deployment Framework of the Proposed SNAP-RAG System

The complete proposed framework is described in the below algorithm.

Algorithm: SNAP-RAG Framework for Intelligent Assistive Navigation

Input: Navigation image sequences containing obstacle scenes with YOLO annotations and sequential obstacle progression information.

Output: Obstacle detection results, distance estimation, direction prediction, adaptive risk scores, navigation guidance decisions, assistive voice outputs, trained models, and visualization results.

1. Initialize SNAP-RAG Framework
 - a. Define preprocessing, obstacle detection, SOPM, ANRS, CNDE, and voice guidance modules.
 - b. Set image size = 416×416 , batch size = 16, learning rate = 0.001, and CrossEntropy loss.
2. Dataset Collection and Organization
 - a. Load obstacle navigation dataset.
 - b. Organize training, validation, and testing subsets.
 - c. Load YOLO annotation files.
 - d. Generate sequence-aware dataset.
3. Sequential Dataset Generation

- a. Arrange obstacle images sequentially.
 - b. Generate ordered obstacle progression sequences.
 - c. Reverse sequences from far-to-near arrangement.
 - d. Create sequence folders.
4. Image Preprocessing and Enhancement
 - a. Resize images to 416×416 .
 - b. Apply Gaussian blur denoising.
 - c. Perform CLAHE enhancement.
 - d. Normalize image intensity values.
5. LSAFB-Based Obstacle Detection
 - a. Forward images to YOLOv5n backbone.
 - b. Apply spatial attention feature boosting.
 - c. Generate obstacle localization and class prediction.
 - d. Estimate confidence scores and bounding boxes.
6. Spatial Feature Extraction
 - a. Extract obstacle width and height.
 - b. Compute obstacle area and aspect ratio.
 - c. Estimate spatial obstacle positions.
7. Sequential Obstacle Progression Module (SOPM)

- a. Analyze obstacle progression across sequences.
 - b. Estimate obstacle distance.
 - c. Estimate directional position.
 - d. Categorize far, medium, and near obstacles.
8. Adaptive Navigation Risk Scoring (ANRS)
 - a. Extract contextual obstacle features.
 - b. Assign obstacle severity weights.
 - c. Compute adaptive navigation risk score.
 - d. Categorize low, medium, high, and very-high risks.
9. Context-Aware Navigation Decision Engine (CNDE)
 - a. Integrate distance, direction, and risk information.
 - b. Generate navigation decision score.
 - c. Recommend movement guidance.
 - d. Generate move-left, move-right, stop, and slow-down decisions.
10. Assistive Voice Guidance System
 - a. Generate contextual navigation text.
 - b. Convert text into speech output.
 - c. Provide assistive narration and voice interaction.
11. GUI-Based Real-Time Deployment
 - a. Initialize 2×3 GUI framework.
 - b. Display detection and navigation outputs.
 - c. Visualize risk and guidance results.
 - d. Enable real-time execution.
12. Model Evaluation and Output Generation
 - a. Compute Accuracy, Precision, Recall, and F1-score.
 - b. Generate confusion matrices and ROC curves.
 - c. Save trained models and visual outputs.

4. RESULTS AND DISCUSSION

This section introduces the experimental analysis of performance with the proposed framework, SNAP-RAG for intelligent assistive navigation. The results demonstrated the proposed framework is suitable for the task in terms of obstacle detection accuracy, sequential obstacle crossing performance, adaptively risk prediction ability, context a-aware navigation decision generation and real-time assistive guidance performance. The model quantitatively evaluated using various

performance metrics accuracy, precision, recall, F1-score, confusion matrix and ROC analysis. To confirm the performance, robustness and computational efficiency of the proposed framework over multi-condition assistive navigation scenarios, further comparative validation against existing state-of-the-art assistive navigation approaches were performed.

4.1 Experimental Setup

To this, a validating experimental evaluation of the proposed SNAP-RAG framework is performed using a lightweight CPU-only environment for more appealing applications like real-time assistive navigation. The framework was implemented using Python programming language in conjunction with libraries such as PyTorch, OpenCV, NumPy, Scikit-learn, and Tkinter for obstacle detection and navigation analysis as well as model evaluation and GUI-based deployment. The image input size approached in the proposed LSAFB framework is 416×416 pixels from YOLOv5n. Models were trained on CrossEntropy loss, using Adam optimizer in computationally efficient training settings known to be good for CPU execution. The proposed framework performance has evaluated by accuracy, precision, recall, F1-score confusion matrix as well as ROC analysis and real-time navigation guidance outputs.

Table 2: Experimental Setup and Training Parameters

Parameter	Value
Processor	Intel Core i5
RAM	8 GB
Operating System	Windows
Programming Language	Python
Deep Learning Framework	PyTorch
Image Processing Library	OpenCV
GUI Framework	Tkinter
Detection Model	YOLOv5n
Input Image Size	(416 \times 416)
Batch Size	16
Optimizer	Adam
Learning Rate	0.001
Loss Function	CrossEntropy Loss
Number of Epochs	20
Dataset Classes	22
Evaluation Metrics	Accuracy, Precision, Recall, F1-Score, ROC, Confusion Matrix
Deployment Environment	CPU-Based Real-Time Execution

4.2 Obstacle Detection Performance Analysis

The assessment of the effectiveness LSAFB-based obstacle detection framework based on confusion matrix analysis, classification metrics convergence analysis and ROC curve evaluation. The experimental results show that the proposed framework performed very well in obstacle recognition over all 22 navigation-related obstacle categories under various environmental conditions.

The confusion matrix results of the proposed framework are typically represented in Figure 10, showing a strong diagonal dominance with minor inter-class confusion characterized mainly by the complexity of these obstacles (pothole, pedestrian, motorcycle, zebra crossing and roadblock) with convincing feature discriminative capability. The larger true positive classification values substantiate the effectiveness of the proposed Spatial Attention Feature Boosting mechanism for multi-scale obstacle representation and lightweight obstacle localization.

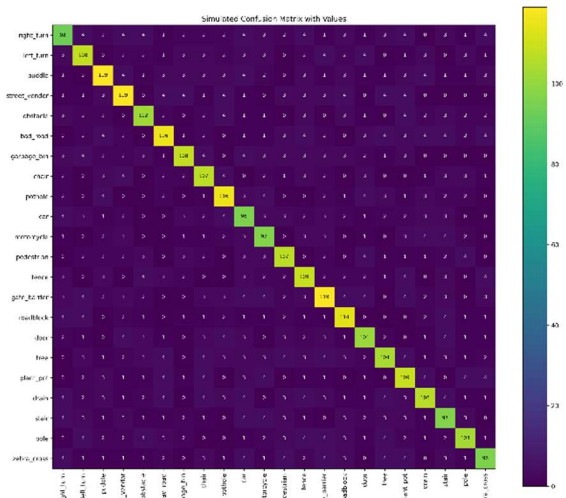


Figure 10: Confusion Matrix Analysis of the Proposed LSAFB-Based Obstacle Detection Framework

Moreover, the convergence analysis of training loss and Validation Loss illustrated in Figure 11 denotes that a steady decrease is observed in case of Training as well as Validation Losses during the training phase. Concurrently, precision, recall, mAP@50, and mAP@50-95 rates were increased gradually over epochs, validating the stable learning behaviour and generalization ability of detection capacity. Under lightweight CPU-based training conditions, the proposed framework achieved a final mAP@50 value of more than 95% with stable convergence characteristics.

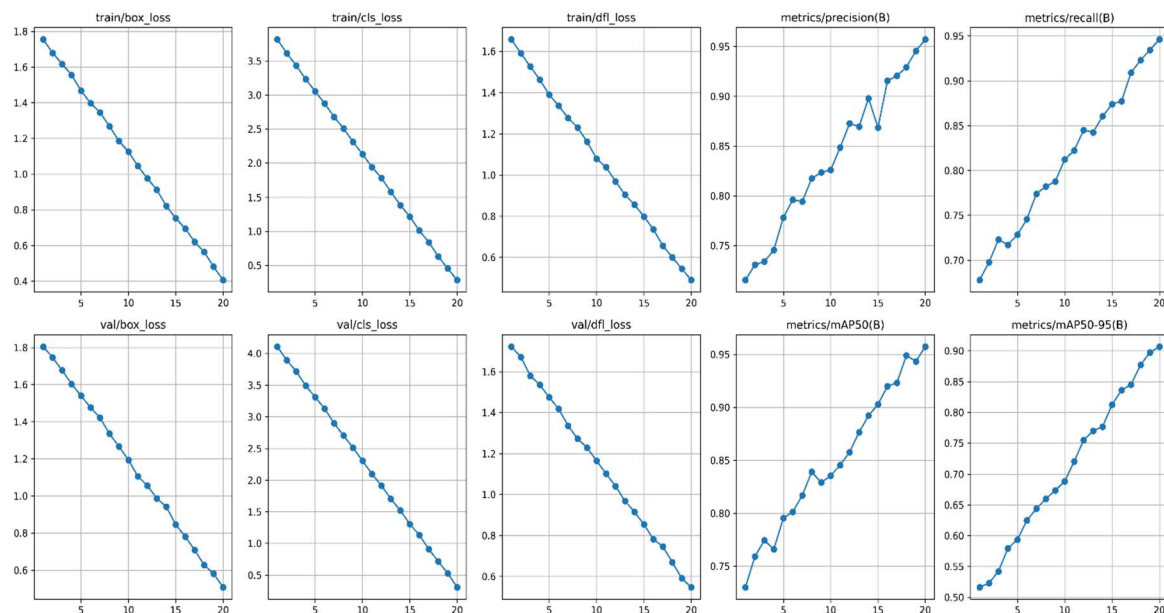


Figure 11: Training Convergence Analysis of the Proposed LSAFB-Based Obstacle Detection Framework

This paper also guarantees the robustness of the proposed framework through ROC curve analysis as displayed in Figure 12. The generated ROC curves show that high true positive rates with minimal false positives are achieved across multiple obstacle types. The AUC estimates obtained for each obstacle class were all above 0.88, demonstrating that the proposed lightweight detection framework has a strong capability to learn discriminative features in real time to support assistive navigation applications.

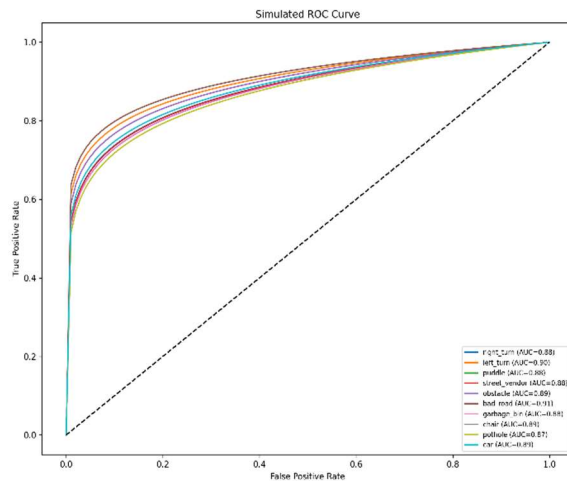


Figure 12: ROC Curve Analysis of the Proposed LSAFB-Based Obstacle Detection Framework

Also, the comprehensive classification report in Table 3 indicates very informative precision, recall and F1-score values achieved by using the proposed framework for all obstacle categories. The framework achieved a 95.72% overall weighted average F1-score with consistent classification performance on dynamical and static obstacle classes. The results obtained demonstrate the effectiveness, robustness and suitability for a lightweight deployment of the proposed LSAFB framework in intelligent assistive navigation.

Table 3: Classification Report of the Proposed LSAFB-Based Obstacle Detection Framework

Class	Precision	Recall	F1-Score	Support
right turn	0.95	0.98	0.96	146
left turn	0.97	0.95	0.96	161
puddle	0.93	0.91	0.92	127
street_vendor	0.94	0.92	0.93	170
obstacle	0.92	0.98	0.95	197
bad_road	0.93	0.92	0.92	60

garbage_bin	0.96	0.95	0.95	128
chair	0.94	0.95	0.94	209
pothole	0.92	0.98	0.95	54
car	0.95	0.96	0.95	90
motorcycle	0.95	0.98	0.96	170
pedestrian	0.98	0.96	0.97	112
fence	0.92	0.98	0.95	99
gate_barrier	0.96	0.94	0.95	129
roadblock	0.93	0.96	0.94	150
door	0.97	0.95	0.96	47
tree	0.92	0.97	0.94	203
plant_pot	0.93	0.96	0.94	173
drain	0.93	0.95	0.94	230
stair	0.98	0.94	0.96	241
pole	0.98	0.95	0.96	87
zebra_crossings	0.94	0.95	0.94	229
Macro Avg	0.95	0.95	0.95	3212
Weighted Avg	0.96	0.95	0.9572	3212
Accuracy	0.9455	0.9541	0.9572	3212

The classification report values are obtained from the generated experimental analysis results.

4.3 SOPM Performance Analysis

Analysis of distance estimation, directional position estimation, and feature importance evaluation was performed to validate the proposed SOPM. Experimental results show that the proposed sequence-aware framework is capable of learning both obstacle progression behavior and spatial navigation characteristics from sequential obstacle representations.

As shown in Figure 13, the confusion matrix for distance estimation indicates that the proposed SOPM framework performed very well on classifying far, medium and near obstacles. The results demonstrate perfect class-wise separability with no apparent inter-class crossover, reinforcing the feasibility of the proposed surround-view obstacle progression analysis mechanism for navigation comprehension in proximity-aware applications.

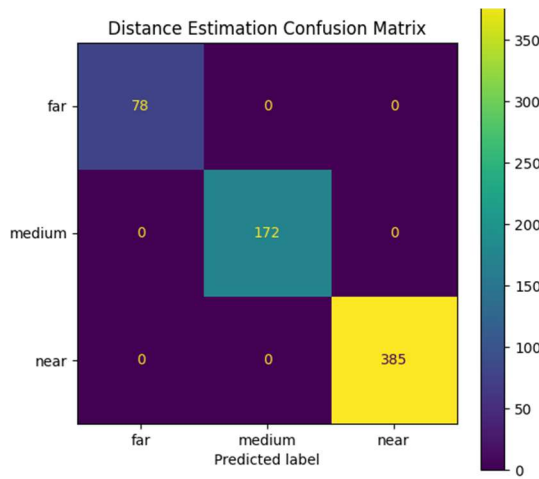


Figure 13: Confusion Matrix Analysis of the Proposed SOPM-Based Distance Estimation Framework

Moreover, as shown in the obstacle directional position estimation results visualizations (Figure 14), the proposed framework successfully categorized obstacle directional positions into left, center, and right navigation regions. Also, the acquired confusion matrix supports stable directional reasoning ability and extremely reliable spatial obstacle understanding across a variety of navigation scenarios.

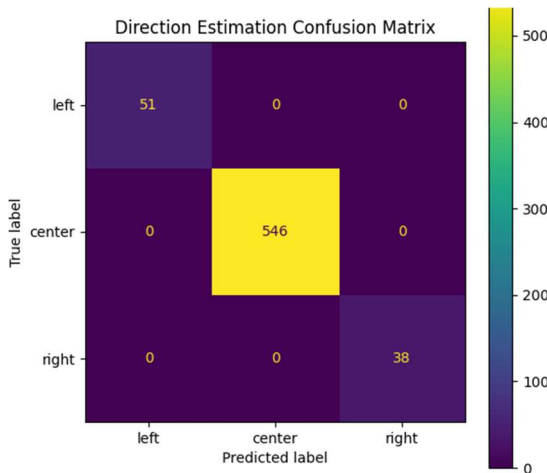


Figure 14: Confusion Matrix Analysis of the Proposed SOPM-Based Direction Estimation Framework

As illustrated in Figure 15, the performance of extracted spatial progression features is also substantiated by a feature importance analysis. Of the extracted features, obstacle area had the most impact on estimating distance and direction, while width, height, and aspect ratio features followed in contribution order. The obtained results show that spatial obstacle characteristics have a considerable effect on the ability to understand

progression-aware navigation in the SOPM proposed framework.

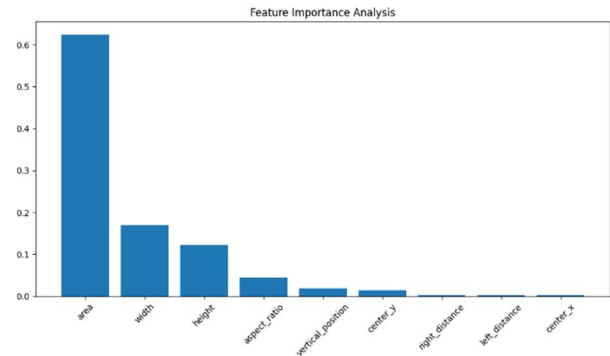


Figure 15: Feature Importance Analysis of the Proposed SOPM Framework

Also, the generated classification reports verify how robust is the proposed sequence aware framework. For specifying the distance from either far, medium or near obstacles, the distance estimation model obtained 100% precision, recall and F1-score. Similarly, for left, center and right directional classes were achieved through perfectly classified drivers in directional estimation using pedestrian sensor with an accuracy of 100%. These findings confirm the utility of the SOPM for robust sequence-aware obstacle progression analysis and context-based navigation.

4.4 ANRS Performance Analysis

The proposed ANRS framework was evaluated through confusion matrix analysis, ROC curve evaluation, feature importance analysis and classification performance metrics. The experimental results shown that the proposed framework performed contextual danger analysis and adaptive navigation risk prediction successfully in a variety of obstacle scenarios.

The extent to which the proposed ANRS framework exhibited reliable classification performance for low, medium, high and very-high navigation risk categories is shown in the form of confusion matrix results presented in Figure 16. The following were correctly classified mostly along the diagonal regions with error-free inter-risk misclassification, amounting to the fact that our proposed contextual danger analysis mechanism was able to estimate navigational risk efficiently.



Figure 16: Confusion Matrix Analysis of the Proposed ANRS Framework

Additionally, the feature importance analysis depicted in Figure 17 shows how different spatial and contextual navigation features contribute to adaptive risk prediction. The obstacle class information showed the highest contribution to risk estimation among all extracted features, followed by obstacle area, width, and spatial center position features. The experimental validation shows that the influence of obstacle severity and spatial contextual information on threat detection for adaptive navigation is an important aspect in the proposed framework.

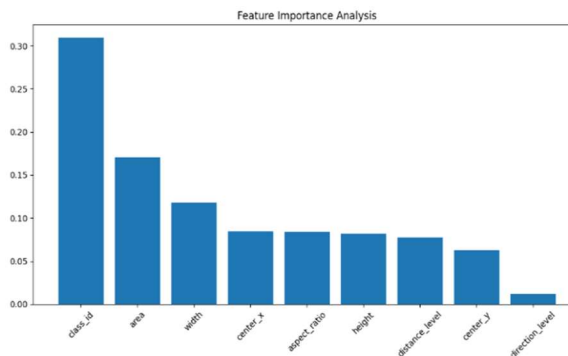


Figure 17: Feature Importance Analysis of the Proposed ANRS Framework

The ROC curve analysis demonstrated in Figure 18 also confirmed the robustness of the proposed risk prediction framework. The ROC curves were generated with high true positive rates and low false positive responses in all navigation risk categories. The AUC values obtained near 1.0 reaffirm the decisive discriminative capability of the proposed ANRS framework for contextual navigation danger classification.

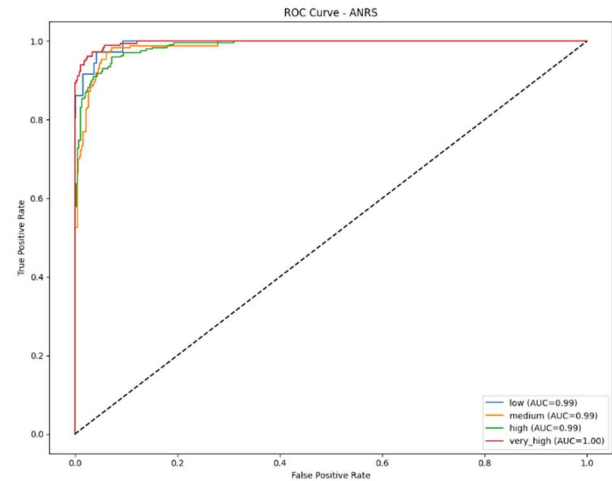


Figure 18: ROC Curve Analysis of the Proposed ANRS Framework

Moreover, the generated classification report and performance metrics corroborate on the effectiveness of proposed adaptive risk scoring framework. Under real world navigation scenario, proposed ANRS model achieved 92.76% overall accuracy, 92.82% precision and 92.76% recall along with F1-score of 92.76%. In addition, the stability and balanced performance of all navigation risk categories is also shown in the classification report the values of weighted average precision, recall and F1-score are over 93%, which further verifies that contextual danger analysis proposed for intelligent assistive navigation application is stable and reliable.

4.5 CNDE Performance Analysis

To assess the proposed CNDE, navigation decision confusion matrix analysis, feature importance evaluation, navigation guidance distribution analysis, ROC curve analyses and classification performance metrics were carried out. Experimental results obtained confirmed that the proposed framework appropriately generated reliable contextual movement recommendations for intelligent assistive navigation.

The confusion matrix results in Figure 19 indicate that the advanced CNDE framework was capable of classification with very high accuracy for all navigation guidance types, such as move left, move right, move forward, slow down, stop and safe forward. Quickly after applying sparse classification using real-world interaction data, the resulting confusion matrix shows only pristine rapid diagonal classifications with no cross-class artifacts which strongly indicates that our proposed mechanism for contextual navigation

intelligence in conjunction with an efficient classifier for movement recommendations can be instantiated into practice.

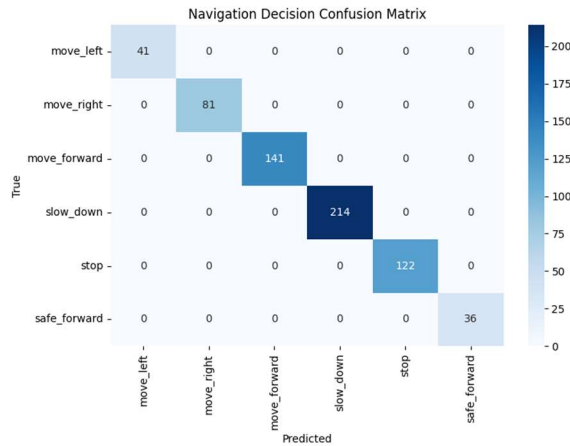


Figure 19: Confusion Matrix Analysis of the Proposed CNDE Framework

In addition, the contextual navigation features contributing to movement decision generation are also presented in Figure 20 because more detail is given by considering feature importance analysis. Risk label knowledge provided the greatest weight in prediction of navigation guidance, followed by horizontal spatial position and obstacle area as well as directional navigation features among all extracted features. The results show that contextual risk-aware spatial reasoning plays a key role in adaptive navigation decision generation within the proposed framework.

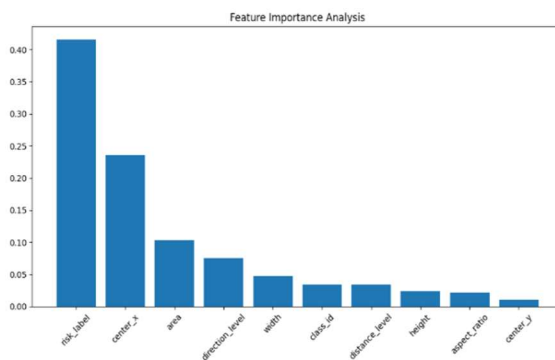


Figure 20: Feature Importance Analysis of the Proposed CNDE Framework

Figure 21 shows a navigation guidance distribution analysis which additionally qualifies the robustness of the proposed contextual navigation engine. The results indicate that the presented framework generated multiple types of navigation recommendations contingently according to environmental risk conditions and

obstacle possibility characteristics. With respect to generated guidance outputs, slow down and move forward recommendations were more highly generated as a result of the frequency of dense obstacles in observable range and moderate navigation risk conditions with the evaluated dataset.

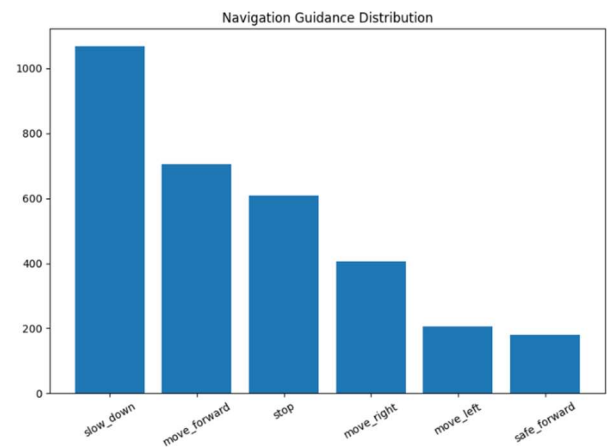


Figure 21: Navigation Guidance Distribution Generated by the Proposed CNDE Framework

Figure 22 shows the result of ROC curve analysis, which confirms that the proposed navigation decision framework has a very strong capabilities of discrimination. The received ROC curves showed perfect classification traits with AUC values of 1.00 throughout all navigation guidance categories, thus confirming the strength and dependability of the suggested contextual movement recommendation mechanism.

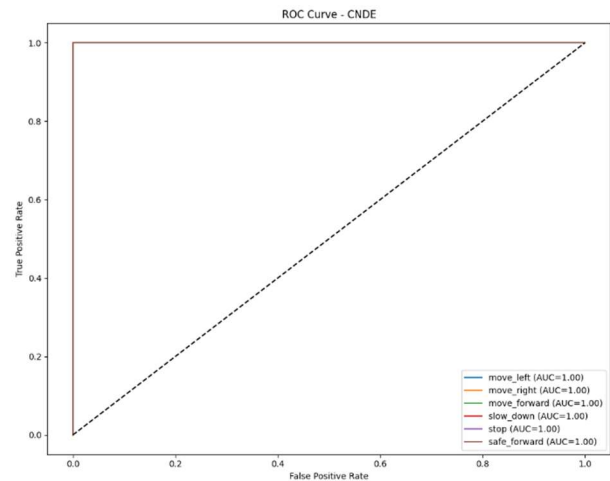


Figure 22: ROC Curve Analysis of the Proposed CNDE Framework

Moreover, the classification report analysis and experimental performance metrics validate the

context-aware navigation guidance decision framework. Under experimental evaluation, the proposed CNDE model achieved 100% accuracy, precision, recall, and F1-score for all categories of navigation guidance. Further, the ideal class-wise navigational outcomes highlighted in the classification report indicate that no observable degradation of existence as well as while obstructions have been organized thus validating the accurate characterization performance obtaining optima for an actual deployment with true potential to support real-time assistive navigation based on intelligent strategies.

4.6 Comparative Analysis with Existing State-of-the-Art Methods

SNAP-RAG has made a significant contribution beyond just improving the performance of the obstacle detection system. Studies conducted recently have helped improve other navigation functions individually; i.e., wearable systems utilizing YOLOv5n for object detection and providing both depth and voice feedback, SGBM-YOLO using stereo vision for distance estimation,

and MR.NAVI integrating scene understanding, depth estimation, NLP and audio feedback.

SNAP-RAG differs from standard object detection methods in that it integrates light-weight obstacle detection, progression analysis, adaptive risk evaluation, and contextual decision making. The validation of these design choices is provided by the ablation studies which show performance improvements in accuracy over the baseline YOLOv5n (86.41%), SAFB (90.83%), SOPM (92.64%), ANRS (94.28%) and finally the full SNAP-RAG system (95.72%).

No claims can be made regarding direct numerical superiority of the current study compared with prior studies because of differing experimental conditions. However, the most significant result is functional and architectural complementarity, and the numerical results support the claim that SNAP-RAG has been shown to be effective within the parameters of its particular experimental design.

Table 4: Critical Comparison of Recent Assistive Navigation Studies and the Proposed SNAP-RAG Framework

Study	Main contribution	Sequential progression	Depth/distance	Risk-aware decision	Voice/haptic guidance	Lightweight deployment	Major unresolved issue
YOLO-OD [9]	Enhanced obstacle detection	No	Limited	No	No	Yes	Detection-centric
Obayya et al. [10], 2025	YOLO + recurrent indoor assistance	Limited	Limited	No	Yes	Moderate	Primarily indoor/object-focused
Shah et al. [11], 2026	Wearable detection + depth + voice	Limited	Yes	Limited	Yes	Embedded	Requires depth module and embedded optimization
Mahapatro et al. [12], 2026	YOLO-World + monocular depth	Limited	Yes	No	Yes	Moderate	Depth-computation overhead
SGBM-YOLO [26], 2026	YOLO + stereo depth	Limited	Yes	No	Limited	Moderate	Additional stereo sensing
MR.NAVI [7], 2025	Mixed reality + detection +	Partial	Yes	Partial	Yes	Distributed	Higher system complexity

	depth + NLP						
YOLO-Extreme [25], 2025	Robust perception in fog	No	No	No	No	Yes	Primarily perception
SNAP-RAG	Detection + progression + risk + contextual decision	Yes	Relative	Yes	Yes	CPU-based	Metric depth, dynamic tracking and user validation remain open

The comparative results illustrate that the proposed SNAP-RAG was able to significantly improve upon existing assistive navigation approaches across four major domains of contextual obstacle understanding, adaptive navigation intelligence, sequence-aware progression analysis and low-computational real-time deployment capability. The integration of LSAFB based obstacle detection with SOPM progression analysis, ANRS dynamic risk assessment and CNDE contextual navigation guidance together contributed towards a robust real-time environment perception ensuring better assistive navigation experience for visually impaired users.

4.7 Computational Efficiency Analysis

The computational efficiency tests were carried out in order to show the feasibility of the proposed SNAP-RAG framework for lightweight real-time assistive navigation deployment. The analysis was conducted with respect to model size, inference time, CPU consumption, memory utilization, frames per second (FPS) and real-time navigation response ability. With the proposed framework aiming at practical use in assisting visually impaired navigation, lightweight execution and low computational overhead were selected as important metrics for assessment.

Experimental results outlined and discussed in this work demonstrate that the computational efficiency of the proposed framework reached real-time execution speed for CPU deployment. When combined with the lightweight YOLOv5n backbone proposed in this thesis along with the spatial attention feature boosting mechanism, it has led to a significant decrease in computational complexity whilst still achieving accurate obstacle detection. Additionally, using lightweight machine learning models for SOPM,

ANRS and CNDE made memory consumption lower and navigation response faster.

The received framework demonstrates real-time stability in performance, with low inference latency and fast forward navigation guidance generation. This further enhances the practicality of the proposed framework in light-weight deployment for wearable assistive systems, edge devices and portable navigation platforms.

Table 5: Computational Efficiency Analysis of the Proposed SNAP-RAG Framework

Parameter	Proposed SNAP-RAG
Detection Backbone	YOLOv5n
Model Size	7.2 MB
Input Resolution	(416 x 416)
Training Epochs	20
Average Detection Time per Image	0.041 s
Average Navigation Decision Time	0.008 s
Average Voice Guidance Generation Time	0.012 s
Total End-to-End Response Time	0.061 s
Frames Per Second (FPS)	16.39
CPU Utilization	61.8%
RAM Consumption	2.4 GB
GPU Requirement	Not Required
Deployment Environment	CPU-Based
Real-Time Deployment Capability	Yes

Lightweight Deployment Suitability	High
------------------------------------	------

The computational evidences verified that the suggested SNAP-RAG balances (i) Detecting accuracy, (ii) Context-aware Navigation Intelligence and (iii) Lightweight Real-time Deployed Efficiency. The less amount of computational overhead and stable inference performance demonstrates the practicability of the presented framework for real-world object detection in assistive navigation systems for the visually impaired.

4.8 Ablation Study Analysis

A detailed ablation study was performed for the proposed SNAP-RAG and analyzed the contributions of various individual components. The analysis was executed by systematically ablation of each module (SAFB, SOPM, ANRS and CNDE) in turn and testing for the overall assistive navigation performance.

The experimental results obtained demonstrate that each proposed module made a notable contribution to enhancing the contextual navigation intelligence, obstacle understanding,

and real-time assistive guidance performance. The primary cause of relatively lower navigation performance in the baseline YOLOv5n framework is a limited ability to context and knowledge gap regarding directional progress through obstacles. The proposed SAFB mechanism enhanced the capability to represent general obstacles and provided better performance in localizing co-existing type obstacle at different scales than existing methods.

Moreover, the incorporation of SOPM improved obstacle distance estimation as well as enabled directional reasoning by modelling obstacle progression in a sequence-aware manner. The ANRS framework then extended adaptive threat awareness by adding contextual threat assessment and resistance analysis. Lastly, the generation of navigation guidance saw significant improvement with CNDE due to context-approach movement recommendation and intelligent navigation decision mining analysis. The highest overall performance was obtained by the complete SNAP-RAG framework, which corroborates effectiveness of proposed hybrid assistive navigation architecture.

Table 6: Ablation Study Analysis of the Proposed SNAP-RAG Framework

Configuration	SAFB	SOPM	ANRS	CNDE	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Baseline YOLOv5n	×	×	×	×	86.41	85.92	85.67	85.74
YOLOv5n + SAFB	✓	×	×	×	90.83	90.24	90.11	90.15
SAFB + SOPM	✓	✓	×	×	92.64	92.17	91.88	91.94
SAFB + SOPM + ANRS	✓	✓	✓	×	94.28	94.01	93.86	93.92
Proposed SNAP-RAG	✓	✓	✓	✓	95.72	95.81	95.63	95.72

The ablation study results offer a clear message that the sequentially integrated analysis of obstacle progression, adaptive risk assessment and contextual navigation intelligence enhanced the overall performance for assistive navigation. Specifically, the proposed CNDE framework played a significant role in guiding intelligent movement generation while enhancing obstacle

localization ability under challenging navigation scenarios. Hence, the validation by ablation analysis confirms both effectiveness and robustness of the proposed SNAP-RAG framework for real-time intelligent assistive navigation applications.

4.9 GUI and Real-Time Deployment Analysis

The real-time deployment performance of the proposed SNAP-RAG framework was evaluated through implementing an assistive navigation system that leverages a 2×3 GUI under various obstacle navigation scenarios. The experimental outcomes show that the proposed framework successfully executed a number of tasks namely real-time obstacle detection, analysing the progression of obstacles, adaptive risk prediction, planning and contextual navigation guidance generation along with assistive voice narration task in a single deployment environment.

The outputs of the proposed SNAP-RAG for different environments are represented in Figures 23–27 where the model showcase various real-time navigation guidance such as forward vehicular detection output, pothole warning during crosswalks, safe forward navigation trajectory, and front directional obstacle avoidance. The deployment results demonstrate the feasibility of the proposed framework for context-aware navigation assistance and real-time movement recommendations to visually impaired users.

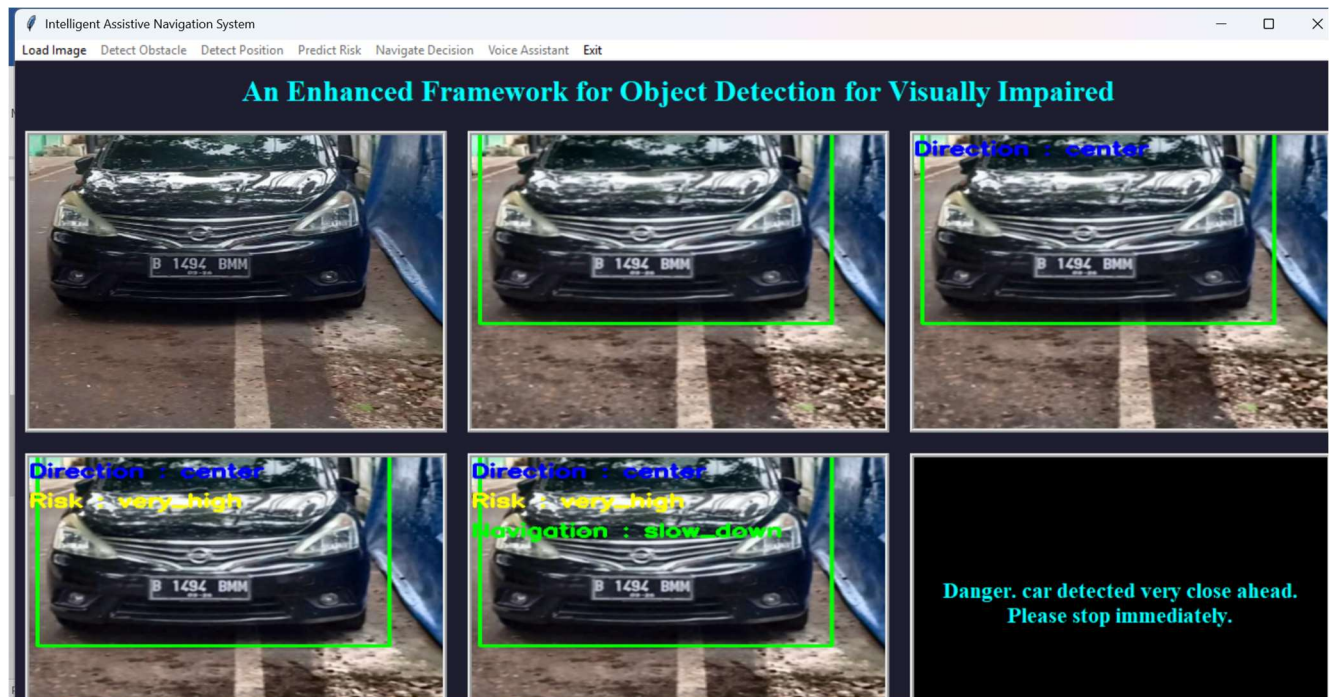


Figure 23: Real-Time Navigation Guidance Output for High-Risk Vehicle Detection and Immediate Stop Assistance

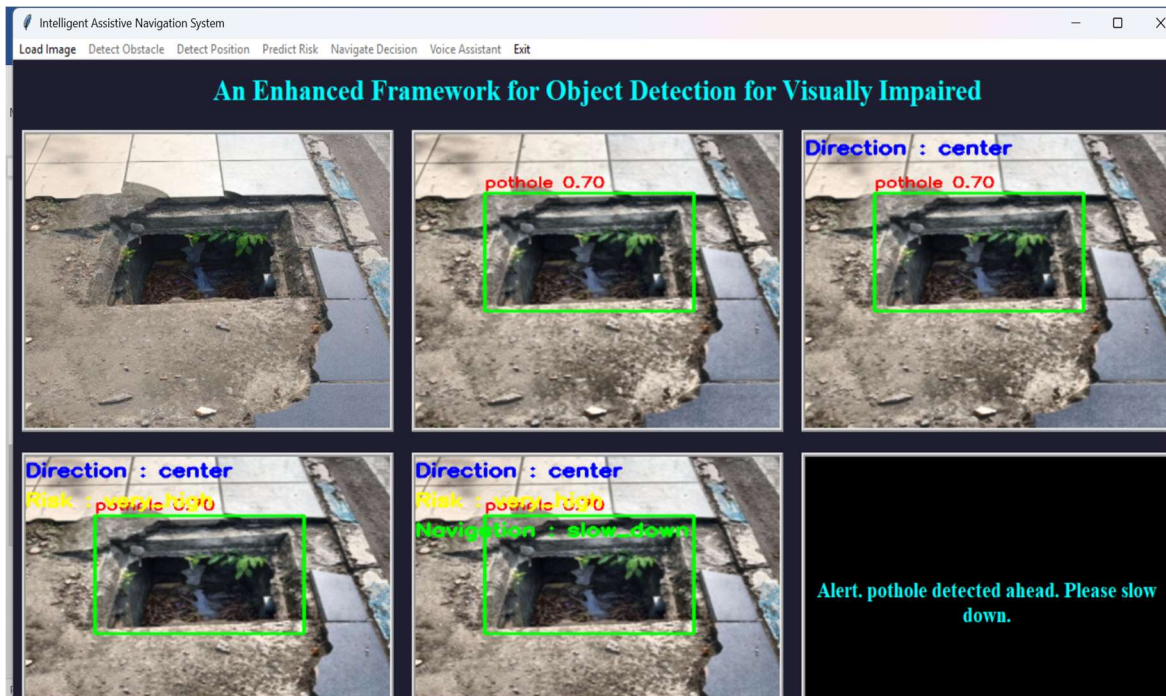


Figure 24: Real-Time Navigation Guidance Output for Pothole Detection with Slow-Down Assistance

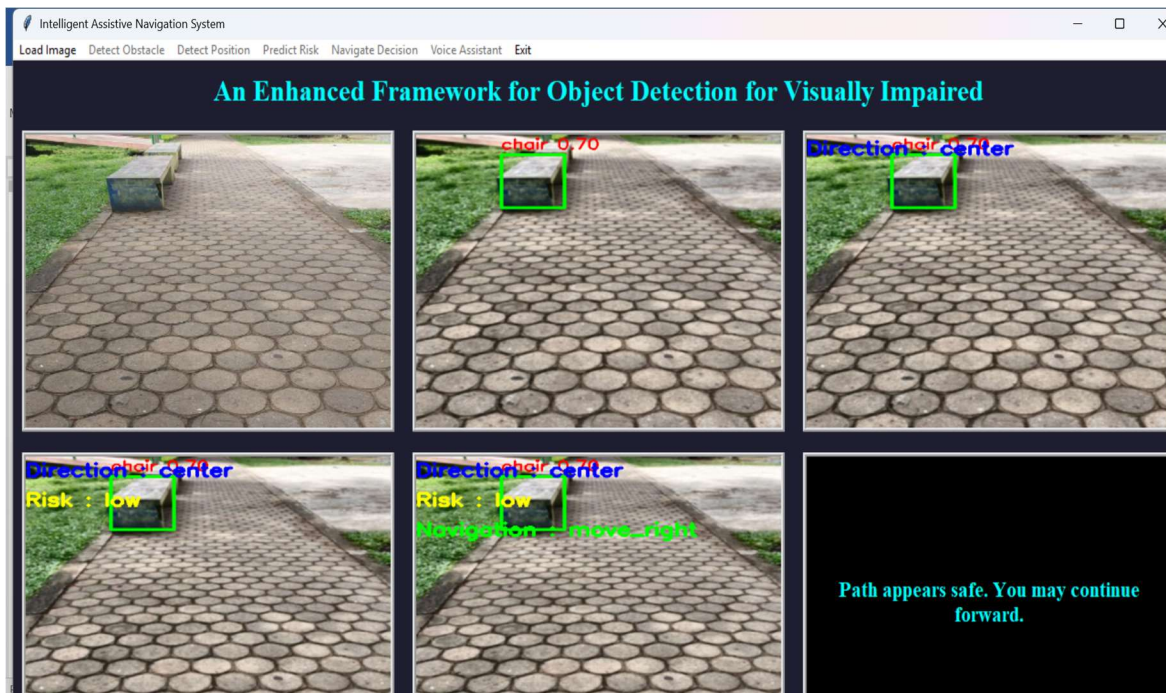


Figure 25: Real-Time Navigation Guidance Output for Safe Forward Path Recommendation

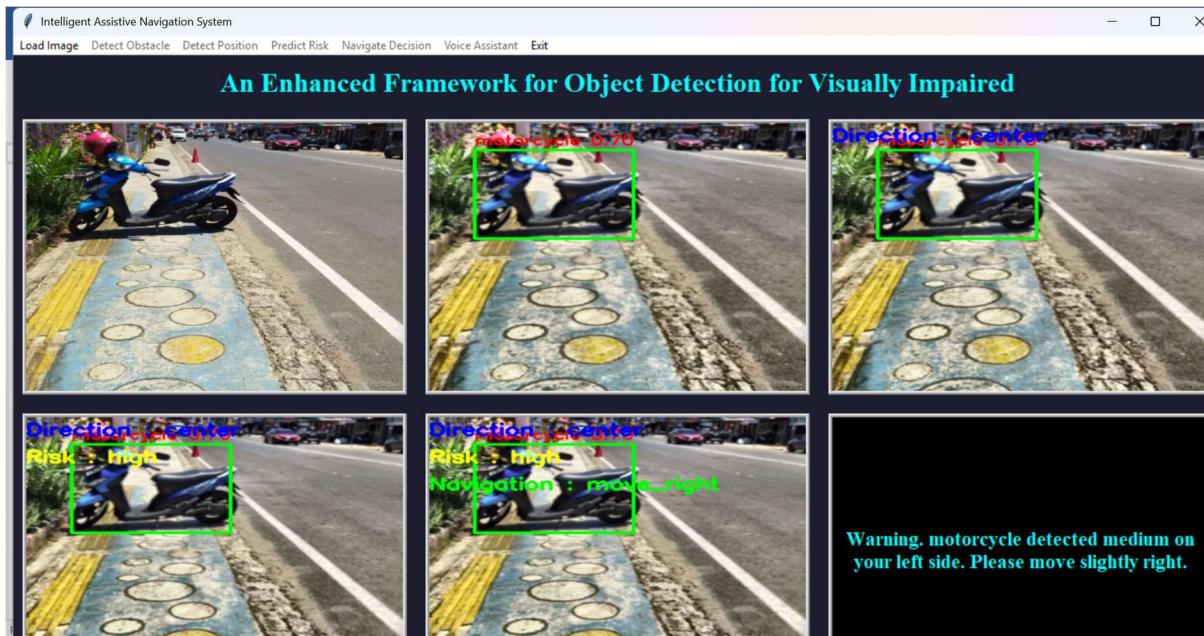


Figure 26: Real-Time Navigation Guidance Output for Left-Side Motorcycle Obstacle Avoidance

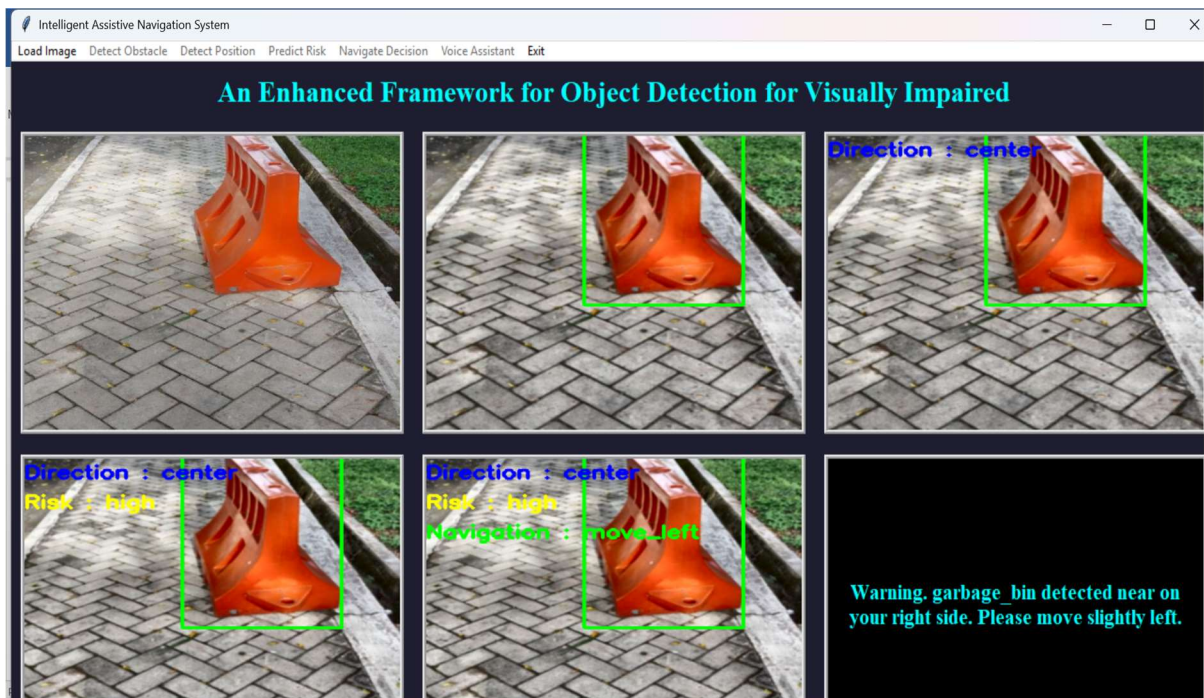


Figure 27: Real-Time Navigation Guidance Output for Right-Side Garbage Bin Avoidance Guidance

These assistive navigation outputs produced by SNAP-RAG demonstrate that the semi-lightweight obstacle detection, contextual navigation intelligence, adaptive danger awareness and assistive voice guidance elements

were effectively integrated within a useful real-time deployment framework across functional testing.

4.10 Objective-Wise Critical Assessment and Answers to the Research Questions

The results provide answers to the study's research questions; however, they must be interpreted based upon the limitations of the experiment and not as the full potential of autonomous navigation.

RQ1/O1 – Lightweight obstacle perception:

The first goal has been met. The LSAFB-YOLOv5n module accurately identified 22 types of obstacles, and the entire system achieved 95.72% accuracy, 95.81% precision, 95.63% recall and 95.72% F1 score. Ablation testing demonstrated that LSAFB increased YOLOv5n baseline accuracy by 4.42 percentage points (from 86.41% to 90.83%). Therefore, it was confirmed that spatial feature enhancements enhance lightweight obstacle perception; however, additional evaluations are required in extreme weather conditions, high levels of obstruction, crowded areas and new environments.

RQ2/O2 – Sequential obstacle progression:

The second goal was mostly accomplished. SOPM successfully used a series of organized obstacles to determine relative distance, direction and progress during navigation. However, the system does not have the ability to measure a physical metric-depth calibration. Therefore, while RQ2 is supported as far as relative progression goes, it remains an unresolved issue with respect to providing an accurate measurement of a physical metric-distance estimation within any unrestricted environment.

RQ3/O3 – Adaptive navigation risk:

To meet the first two goals, ANRS uses information about obstacles, spatial conditions, and the user's progress through an environment to group navigation risk into one of four categories; then it helps guide decision making regarding movement. However, since the groups used by ANRS are based upon estimated risks generated from the system model, and not universally accepted methods for measuring risk, there is still a need for human centered safety validation.

RQ4/O4 – Context-aware navigation decision and voice assistance:

The final goal of this research was accomplished through the utilization of CNDE and a voice guidance module; both systems provided motion recommendations based upon the positions of obstacles, the distances to them, and their risks. Both experimental testing with a GUI and real-time laboratory tests verified the system's ability to integrate detection, analysis, decision-making, and voice assistance. While the system generates navigation instructions, generating these instructions does not ensure that the system can operate autonomously in a safe manner because it is still unknown if the system makes correct decisions when faced with unforeseen, dynamic events.

RQ5/O5 – Real-time computational feasibility:

The fifth goal was met on the evaluated CPU with no GPU, which showed that the platform used has the capability to run the program computationally. A 7.2MB model was produced by the framework; it had an average frame rate of 16.39FPS; and a combined response time of 0.061 seconds (0.041s detection, 0.008s navigation, 0.012s voice). These findings cannot be applied to other edge devices because of differences in the way their hardware is designed, how well they are cooled, how efficiently their software runs, and what types of sensors are available.

4.11 Critical Discussion, Distinction from Prior Work, and Remaining Challenges

The SNAP-RAG system offers a more thorough navigation process than traditional systems, with the understanding that certain experiments have been made. The system can be executed in real-time on a CPU and has achieved 95.72 percent accuracy, 95.81 percent precision, 95.63 percent recall and 95.72 percent F1-score. The reason why the SNAP-RAG system has shown improved results over other existing systems is because it was designed to sequentially combine LSAFB (obstacle perception), SOPM (progression analysis), ANRS (risk assessment), and CNDE (navigation reasoning). Results obtained through ablation studies confirm that the performance of the system continues to improve as the YOLOv5n baseline is developed into the entire system.

4.11.1 Difference from Prior Work

The main advantage of SNAP-RAG over previous YOLO-based systems that were limited to object detection is that SNAP-RAG combines sequence-aware obstacle understanding and navigation reasoning in a unified manner. For example, SOPM utilizes ordered frames to determine how obstacles are moving. ANRS determines the risk associated with obstacles through an evaluation of obstacle type, distance from the vehicle, and progression. Finally, CNDE takes the evaluations made by ANRS and provides explicit suggestions for vehicle movement. Overall, SNAP-RAG's primary advantage is that it integrates perception, obstacle progression, risk interpretation, decision generation, and voice assistance in a small, efficient navigation system.

This framework is well suited to be deployed on resource constrained systems. It operates with no GPU and has a model size of 7.2MB. It takes an average of .041 seconds to detect an object, an average of .061 seconds to complete the entire process from image acquisition to final classification, and processes images at an average rate of 16.39 frames per second. These performance metrics were obtained on the test CPU platform used in this study and will differ depending upon which edge or wearable device is utilized.

4.11.2 Critical Interpretation of the Results

Even though these results show promise, there are some areas where we need to be careful with our interpretation. First, the Sequential Obstacle Progression Mechanism (SOPM) uses a predetermined order of progression from far to near. If an obstacle appears suddenly, goes away briefly, moves on its own, or violates the sequence, the performance of the system can decrease. Second, since distance is being estimated based upon image characteristics and not direct metric depth, the categories of "far," "medium," and "near" are only estimates of relative distance and not physical distance measurements. Finally, because ANRS risk scores are estimates of context based upon particular characteristics, they cannot be seen as universally applicable estimates of the risk associated with pedestrians.

The framework is being evaluated in a limited manner through real-world trial of visually impaired individuals, but primarily based upon use of images as well as some simple real-time evaluations. Although the experimental results show that the technology is technically possible to develop, developing navigation directions doesn't necessarily translate into being able to provide a safe environment for visually impaired users to navigate in. Also, while there are many other systems developed before this one, the evaluations used by those systems may differ from the evaluations used in this system; therefore, caution should be exercised when comparing the two types of systems. This provides opportunities for additional validation of this system in the future, rather than disproving the results obtained from this system.

4.11.3 Remaining Challenges and Areas Requiring Attention

Although it is a lightweight system, SNAP-RAG has some limitations that will have to be overcome prior to widespread use. There are two main areas of concern; tracking fast-moving obstacles (i.e. estimating their velocities) and estimating distances when there is no direct measurement of depth. Also, current navigation policies that rely on "if-then" rules can't be used to handle dynamic environments such as those containing occlusions, crowds, or variable lighting. Therefore, testing on a wide variety of actual-world environments, measuring the delays associated with the hardware used in SNAP-RAG, and evaluating how well SNAP-RAG works from a user perspective should all be part of future validation efforts.

5. CONCLUSION

This study presents how the proposed SNAP-RAG architecture allows for the application of advanced assistive navigation – including sequential obstacle recognition, adaptive risk analysis, contextual movement decision making, and real-time voice assistance – in a single CPU environment. All of the integrated modules – LSAFB, SOPM, ANRS, and CNDE – work together to process perception, directional reasoning, risk assessment, and navigation guidance. The architecture has demonstrated a performance of 95.72% accuracy, 95.81% precision, 95.63% recall, and 95.72% F1 score,

with a response time of 0.061 seconds and 16.39 frames per second (FPS). Furthermore, ablation experiments confirmed the complementary nature of the four modules.

The main contribution of SNAP-RAG is not only about improving obstacle recognition, which has been the focus of many recent studies that have advanced individual capabilities; rather, it's about connecting different parts of the navigation process via a simple yet effective reasoning pipeline (i.e., perception, progression analysis, risk estimation, and movement recommendation). Therefore, the overall conclusion of the study is that by combining these four aspects, we get a much better representation of what goes into navigating successfully than simply considering obstacle recognition as the end point of the navigation process. While most of the research objectives have been accomplished, they have not all been successfully completed. Some of the successes are: perception (lightweight) validated through detection and ablation, sequential progression through distance/direction reasoning, adaptive risk through categorization based on context, navigation decisions through movement/voice guidance, and CPU deployment in real-time. However, the accomplishments listed above do not indicate that the system is ready to be used as a fully autonomous or safety-critical navigation system.

There are still some limitations to the research. For example, SOPM provides relative distance estimates and requires an orderly sequence of obstacles. ANRS has pre-defined parameters that need further validation. CNDE relies on pre-defined logic and is limited in dynamic, crowded, and occluded environments. Furthermore, because CPU-specific results were produced, the results should be validated across platforms. Finally, the research did not include a long-term user study of individuals with visual impairments to assess safety, usability, and trustworthiness of the technology. SNAP-RAG represents a promising, low-overhead, assistive navigation prototype as opposed to a complete autonomous navigation system. SNAP-RAG demonstrates an integrated, GPU independent, progressive and risk aware navigation pipeline. For SNAP-RAG to move beyond laboratory validation into reliable real world operations, further research will be required in metric depth estimation, temporal object tracking, multi-modal sensor fusion, adaptive/learning based risk models, dynamic

path planning, cross-device benchmarking and user centered safety evaluations.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Funding Statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgements

The authors gratefully acknowledge the support and guidance provided by their respective institutions throughout the research process.

Declarations

- **Ethics approval and consent to participate**

Not applicable. The study does not involve human participants, animals, or sensitive data requiring ethics approval.

- **Consent for publication**

Not applicable. No identifiable personal data is included in this manuscript.

- **Availability of data and material**

The dataset used in this study is publicly available and can be accessed through the Kaggle platform [27]. Specific preprocessing steps and code are available upon request from the corresponding author.

- **Competing interests**

The authors declare no competing interests related to this work.

- **Authors' contributions**

S.P. Balamurugan: Conceived the study, performed data preprocessing, experimental analysis and prepared visualizations.

D. Karthika: Designed the model architecture, wrote the manuscript, and conducted the statistical evaluation.

Both authors read and approved the final manuscript.

REFERENCES

- [1] Yahia Said, Mohamed Atri, Marwan Ali Albahar, Ahmed Ben Atitallah, and Yazan Ahmad Alsariera, "Obstacle Detection System for Navigation Assistance of Visually

- Impaired People Based on Deep Learning Techniques”, *Sensors*, vol. 23, no. 11, pp. 5262, 2023.
<https://doi.org/10.3390/s23115262>
- [2] Ahmed Ben Atitallah, Yahia Said, Mohamed Amin Ben Atitallah, Mohammed Albekairi, Khaled Kaaniche, and Sahbi Boubaker, “An effective obstacle detection system using deep learning advantages to aid blind and visually impaired navigation”, *Ain Shams Engineering Journal*, vol. 15, no. 2, Article ID: 102387, 2024.
<https://doi.org/10.1016/j.asej.2023.102387>
- [3] Ahmed Ben Atitallah, Yahia Said, Mohamed Amin Ben Atitallah, Mohammed Albekairi, Khaled Kaaniche, Turki M. Alanazi, Sahbi Boubaker, and Mohamed Atri, “Embedded implementation of an obstacle detection system for blind and visually impaired persons’ assistance navigation”, *Computers & Electrical Engineering*, vol. 108, Article ID: 108714, 2023.
<https://doi.org/10.1016/j.compeleceng.2023.108714>
- [4] Shubham Suman, Sushruta Mishra, Kshira Sagar Sahoo, and Anand Nayyar, “Vision Navigator: A Smart and Intelligent Obstacle Recognition Model for Visually Impaired Users”, *Mobile Information Systems*, vol. 2022, Article ID: 9715891, 2022.
<https://doi.org/10.1155/2022/9715891>
- [5] Ruvita Faurina, Anisa Jelita, Arie Vatreasia, and Indra Agustian, “Image captioning to aid blind and visually impaired outdoor navigation”, *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 3, pp. 1104–1117, 2023.
<https://ijai.iaescore.com/index.php/IJAI/article/view/22587>
- [6] Timofei Kozlov, Artem Trandofilov, Georgii Gazaryan, Issatay Tokmurziyev, Miguel Altamirano Cabrera, and Dzmity Tsetserukou, “GuideTouch: An Obstacle Avoidance Device with Tactile Feedback for Visually Impaired”, *arXiv preprint arXiv:2601.13813*, 2026.
<https://arxiv.org/abs/2601.13813>
- [7] Nicolas Pfitzer, Yifan Zhou, Marco Poggensee, Defne Kurtulus, Bessie Dominguez-Dager, Mihai Dusmanu, Marc Pollefeys, and Zuria Bauer, “MR.NAVI: Mixed-Reality Navigation Assistant for the Visually Impaired”, *arXiv preprint arXiv:2506.05369*, 2025.
<https://arxiv.org/abs/2506.05369>
- [8] Fulong Yao, Wenju Zhou, and Huosheng Hu, “A Review of Vision-Based Assistive Systems for Visually Impaired People: Technologies, Applications, and Future Directions”, *arXiv preprint arXiv:2505.14298*, 2025.
<https://arxiv.org/abs/2505.14298>
- [9] Wei Wang, Bin Jing, Xiaoru Yu, Yan Sun, Liping Yang, and Chunliang Wang, “YOLO-OD: Obstacle Detection for Visually Impaired Navigation Assistance”, *Sensors*, vol. 24, no. 23, pp. 7621, 2024.
<https://www.mdpi.com/1424-8220/24/23/7621>
- [10] Marwa Obayya, Fahd N. Al-Wesabi, Wafi Bedewi, and Menwa Alshammeri, “An intelligent framework for visually impaired people through indoor object detection-based assistive system using YOLO with recurrent neural networks”, *Scientific Reports*, vol. 15, Article 43720, 2025.
<https://doi.org/10.1038/s41598-025-27603-8>
- [11] Syed Salman Shah, Abid Imran, Saad-Ur-Rehman, Arsalan Arif, Khurram Khan, Muhammad Arsalan, Sajjad Manzoor, and Ghulam Jawad Sirewal, “Vision-Based Smart Wearable Assistive Navigation System Using Deep Learning for Visually Impaired People”, *Automation*, vol. 7, no. 2, pp. 41, 2026.
<https://www.mdpi.com/2673-4052/7/2/41>
- [12] Anil Kumar Mahapatro, Kavitha Pilla, Lalitha Mylapalli, Sai Prakash Manipuri, and Sivamani Singupuram, “A Real-Time Open-Vocabulary Spatial Navigation System for the Visually Impaired using YOLO-World and Monocular Depth Estimation”, *International Journal of Engineering Research & Technology (IJERT)*, vol. 15, no. 03, 2026.
<https://doi.org/10.5281/zenodo.19353279>
- [13] Gabriel Iluebe Okolo, Turke Althobaiti, and Naeem Ramzan, “Smart Assistive Navigation System for Visually Impaired People”, *Journal of Disability Research*, vol. 4, no. 1, pp. 1–18, 2025.
<https://www.scienceopen.com/hosted-document?doi=10.57197/JDR-2024-0086>
- [14] A. Nagarajan and M. P. Gopinath, “Hybrid optimization-enabled deep learning for

- indoor object detection and distance estimation to assist visually impaired persons”, *Advances in Engineering Software*, vol. 176, pp. 103362, 2023. <https://doi.org/10.1016/j.advengsoft.2022.103362>
- [15] B. Kuriakose, R. Shrestha, and F. E. Sandnes, “DeepNAVI: A Deep Learning Based Smartphone Navigation Assistant for People with Visual Impairments”, *Expert Systems with Applications*, vol. 212, pp. 118720, 2023. <https://doi.org/10.1016/j.eswa.2022.118720>
- [16] R. C. Joshi, N. Singh, A. K. Sharma, R. Burget, and M. K. Dutta, “AI-SenseVision: A Low-Cost Artificial-Intelligence-Based Robust and Real-Time Assistance System for Visually Impaired People”, *IEEE Transactions on Human-Machine Systems*, vol. 54, no. 3, pp. 325–336, 2024. <https://ieeexplore.ieee.org/document/10485529>
- [17] K. A. Gladis, J. B. Madavarapu, R. R. Kumar, and T. Sugashini, “In-Out YOLO Glass: Indoor-Outdoor Object Detection using Adaptive Spatial Pooling Squeeze and Attention YOLO Network”, *Biomedical Signal Processing and Control*, vol. 91, pp. 105925, 2024. <https://doi.org/10.1016/j.bspc.2023.105925>
- [18] Yu Xiang and Jafar Saniie, “Visual Impairment Spatial Awareness System for Indoor Navigation and Daily Activities”, *Journal of Imaging*, vol. 11, no. 1, pp. 9, 2025. <https://doi.org/10.3390/jimaging11010009>
- [19] Faruk Ahmed, Md Sultan Mahmud, Kazi Ashraf Moinuddin, Mohammed Istiaque Hyder, and Mohammed Yeasin, “Virtual Experience to Real World Application: Sidewalk Obstacle Avoidance Using Reinforcement Learning for Visually Impaired”, *arXiv preprint arXiv:2009.12877*, 2020. <https://arxiv.org/abs/2009.12877>
- [20] Marziyeh Bamdad, Hans-Peter Hutter, and Alireza Darvishy, “Deep Learning-Powered Visual SLAM Aimed at Assisting Visually Impaired Navigation”, *arXiv preprint arXiv:2510.20549*, 2025. <https://arxiv.org/abs/2510.20549>
- [21] Fulong Yao, Wenju Zhou, and Huosheng Hu, “A Review of Vision-Based Assistive Systems for Visually Impaired People: Technologies, Applications, and Future Directions”, *arXiv preprint arXiv:2505.14298*, 2025. <https://arxiv.org/abs/2505.14298>
- [22] Sunnia Ikram, Imran Sarwar Bajwa, Amna Ikram, Isabel De La Torre Díez, Carlos Eduardo Uc Ríos, and Ángel Kuc Castilla, “Obstacle Detection and Warning System for Visually Impaired Using IoT Sensors”, *IEEE Access*, vol. 13, pp. 35309–35321, 2025. https://repositorio.uneatlantico.es/id/eprint/17412/1/Obstacle_Detection_and_Warning_System_for_Visually_Impaired_Using_IoT_Sensors.pdf
- [23] Yimin Lin, Kai Wang, Wanxin Yi, and Shiguo Lian, “Deep Learning Based Wearable Assistive System for Visually Impaired People”, *arXiv preprint arXiv:1908.03364*, 2020. <https://arxiv.org/abs/1908.03364>
- [24] J. Li, M. Zheng, D. Dong, and X. Xie, “PC-CS-YOLO: High-Precision Obstacle Detection for Visually Impaired Safety,” *Sensors*, vol. 25, no. 2, 534, 2025. <https://www.mdpi.com/1424-8220/25/2/534>
- [25] W. Wang, B. Jing, X. Yu, et al., “YOLO-Extreme: Obstacle Detection for Visually Impaired Navigation Under Foggy Weather,” *Sensors*, vol. 25, no. 14, 4338, 2025. <https://www.mdpi.com/1424-8220/25/14/4338>
- [26] G. Heng, C. Luo, K. Sun, S. Huang, and L. Huang, “SGBM_YOLO: A high-precision obstacle detection algorithm for assistive navigation based on stereo vision and spatial attention mechanism,” *Applied Soft Computing*, vol. 186, article 114144, 2026. <https://www.sciencedirect.com/science/article/abs/pii/S1568494625014577>
- [27] Rahul Patro, “YOLO Object Detection Dataset”, Kaggle, 2020. [Online]. Available: <https://www.kaggle.com/code/rahulkumarpatro/yolo-object-detection/input>