APEX ACADEMIA PRESS
ELEGANCE ON EVERY RESEARCH

International Journal of Engineering & Tech Development
ISSN: 3067-669X / Open Access / IJETD_Y2F2A001
Apex Academia Press / https://doi.org/10.63665/ijetd-y2f2a001

Original Article

An Intelligent Assistive Navigation Guide System for Blind and Visually Impaired Individuals Using Deep Learning Techniques

Osita Miracle Nwakeze

Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University Uli, Anambra State.

Abstract

To satisfy the need for more advanced and reliable navigation support systems, a growing number of visually impaired people is faced with the problem of navigation in various environments, and conventional assistive devices, like white canes or common electronic aids, are not sufficient. This work was aimed at creating a smart real-time assistive system to detect the surrounding obstacles, estimate its distance from the user, and give immediate audio information to help to move safely and independently. Developed with a hybrid approach of Agile and Behaviour-Driven Development, the system is flexible, progressive, and user-centric, supporting the user through ongoing enhancements. The choice of the object detection model was made, in accordance with the requirements, on the basis of speed and high detection accuracy of deep learning, which is YOLOv8. In order to perform better in real challenging world, an anti-occlusion model was added to better recognize the partially hidden objects. An algorithm called Distance Estimation Algorithm was implemented to estimate the proximity of an object and a Kalman filter was used for dynamic tracking of an object. The results of the detection were then displayed in speech by using a Text-to-Speech module. A data set of 329,067 images was used for the training of the model, which was created by fusing images from primary image capture with the ZED 2i with images from the Microsoft COCO dataset. Experimental results demonstrated that the system had the precision of 97.4%, recall of 95.2% and the mAP@0.5 of 86.9%. The system was tested in practice, and the results were satisfactory, with accurate detection of obstacles, estimation of distance, and real-time audio notification. The authors of the study have concluded that the created system is an effective and intelligent navigation tool that can increase mobility, safety and independence of visually impaired people.

Keywords

Assistive Navigation System, Visual Impairment, Deep Learning; YOLOv8, Obstacle Detection.

Article History

Received: 07.04.2026

Accepted: 18.04.2026

Published: 24.04.2026

1. Introduction

Nature's miracle of sight enables us to navigate, interpret, and connect with the world. Yet, for millions of visually impaired individuals, this vital sense remains a challenge, demanding innovative solutions to bridge the gap between vision and interaction. According to Anna (2024), in 2020, around 1.1 billion people worldwide had vision impairment, with 43 million of them living with complete blindness. This number has been projected to increase massively in 2050, with an estimated 1.8 billion individuals experiencing impaired vision, with an estimated 100 million people with complete blindness worldwide (Burhan *et al.* (2023). In addition, Anna (2024) predicted that in 2050, one out of every six children will have issues of impaired vision.

The World Health Organization (WHO, 2023) reported that the global financial burden as a result of vision impairment includes an annual productivity cost of 411 billion USD. Beyond this financial cost, other impacts are social isolation, accidents, depression, anxiety, and dependence on other people's guidance for mobility and daily activities. This underscores the need for advance intervention to address the rising prevalence and impacts of vision impairment.

Among the interventions, assistive technologies (AT) for the visually impaired have continued to gain increased momentum in the scientific community, with the market estimated to reach 10.68 billion USD in 2030 from 5.29 billion USD in 2023 (The Business Research Company, 2023). ATs are systems designed to allow people with impaired vision to manage various aspects of their lives. The primary aim of ATs is to enhance visual perception and promote independence and means of navigating more effectively. Some of these ATs are automatic blind guide walking sticks,

electronic touch, Braille systems, speech-to-text software, and smart eyeglasses for impaired vision. While these systems have several benefits already mentioned, they have certain weaknesses, like the impact on health due to vibration, high cost, limited detection of a few obstacles, limited consideration for the completely blind, and general reliability issues.

Recently, computers through the power of deep learning (DL) have resonated as a promising technology to facilitate the modeling of human computer interaction based-AT for the blind. DL is a type of artificial intelligence technique developed from Convolutional Neural Network (CNN) to solve obstacle recognition and avoidance problems. In the context of the vision-impaired users, DL can be trained to detect and recognize obstacles, then notify the user through speech for avoidance. Ebere et al. (2024) grouped obstacle detection models into one-stage and two-stage detectors. The one-stage detectors are algorithms such as You Can Only Look Once (YOLO) (Oluwaseyi et al., 2023) and short single multi box detector (Li et al., 2024), while the two-stage detectors are Recurrent-CNN (R-CNN), Fast-RCNN, and Mask CNN (Bochkvoskiy et al., 2020). While these detectors have been applied in different applications for object detection through computer vision, and Bochkvoskiy et al. (2020) revealed that YOLO is the best for real-time object detection problems. The reasons include high accuracy, speed, fast recognition, and the ability to label classified objects, thus making it the most suitable for this project.

YOLO has different versions, which range from YOLO-V1 to currently YOLOV-10, and have been applied in literature (Rath et al., 2023; Salam et al., 2021; Zhelin et al., 2020) for object detection with several successes recorded, both in terms of accuracy and speed; however, several weaknesses occur due to occlusion of objects and external environments, thus presenting a major challenge that may impact its suitability as the object detection classifier in the human-computer interactive system for blind navigation. In addition, even though YOLOV has been identified in literature as the best object detection classifier at this moment, there is limited application of the model in human-computer interactive systems for impaired vision. Therefore, this thesis proposes designing an assistive navigation guide model system for the blind using deep learning techniques. The major contribution of the study will include adapting YOLOV algorithms to correctly classify obstacles and then notify the user to prevent an accident.

2. Methodology

The methodology for the design of an assistive navigation guide model for the blind using deep learning techniques is hybrid approach (Agile + Behaviour-Driven Development). Agile enables flexibility in development, allowing the system to evolve through short, iterative cycles, where feedback from blind users can be incorporated quickly. Since real-world conditions and user needs can vary, this iterative approach ensures continuous improvement and refinement of the system, making it adaptable to diverse environments and specific requirements. Moreover, agile supports collaboration among developers, designers, and users, allowing for faster identification and resolution of issues.

Behaviour-Driven Development (BDD) complements Agile by emphasizing the behavior of the system in terms of user scenarios, making it easier to define clear, testable requirements that are aligned with the blind users' needs. For instance, test cases like "Given an obstacle detected, the system provides audio feedback with the distance and direction" ensure that the system's functionality is always tested in real-world scenarios. This approach bridges the gap between technical development and user expectations, ensuring that the system's design and output are intuitive and practical for blind users. Together, this hybrid methodology ensures that the system is flexible, user-centered, and rigorously tested, ensuring an effective, practical, and user-friendly navigation aid for blind individuals.

A. Data collection

This section discussed the data collection process. The data used for this work considered both indoor and outdoor data. The indoor data are chair and table, while the outdoor objects are cars, tri-cycles, and door. The instrument for data collection is ZED2i HD camera. The primary data collected is tricycle with 3013 images. The secondary data used is the Microsoft COCO dataset. The COCO dataset is made of 328,000 images of everyday objects, representing 91 different objects which include person, chair, doors, and cars with a total of 2.5 million labeled instances. The overall sample size of objects collected are 329067 images. Collectively, these were used to model the new dataset which contained both indoor and outdoor objects. Figure 1 presents the test data sample.

Fig-1: Sample person data (Author, 2025)

Fig-2: Sample doors (Source: COCO)

Fig-3: Sample cars (Source: COCO)

B. The Proposed YOLO Model

YOLOv8 (You Only Look Once version 8) represents the latest iteration of the YOLO object detection and image segmentation model series. The YOLOv8 series offers a range of models to cater to diverse scenarios and requirements. Among them, YOLOv8n stands out as the most computationally efficient and fastest in inference speed, making it particularly suitable for deployment on resource-constrained devices. Its structure is shown in Figure 4. It consists of three components: Backbone, Neck, and Head. The Backbone module incorporates Conv, C2f, and Spatial Pyramid Pooling Fast (SPPF) components, which, respectively, enhance nonlinear expression, feature propagation, and multi-scale processing capabilities. The Neck part employs a Path Aggregation Network (PAN) structure, combining the Feature Pyramid Network (FPN) to achieve efficient feature fusion. In the Head module, YOLOv8n uses a Decoupled Head design to separately handle localization and classification tasks, improving prediction accuracy and efficiency, simplifying the model structure, and enhancing computational speed. Regarding the loss function, YOLOv8n employs Complete Intersection over Union loss (CIoU) combining IoU and the distance between central points to measure more accurately the difference between the predicted and ground truth boxes, thereby improving localization accuracy.

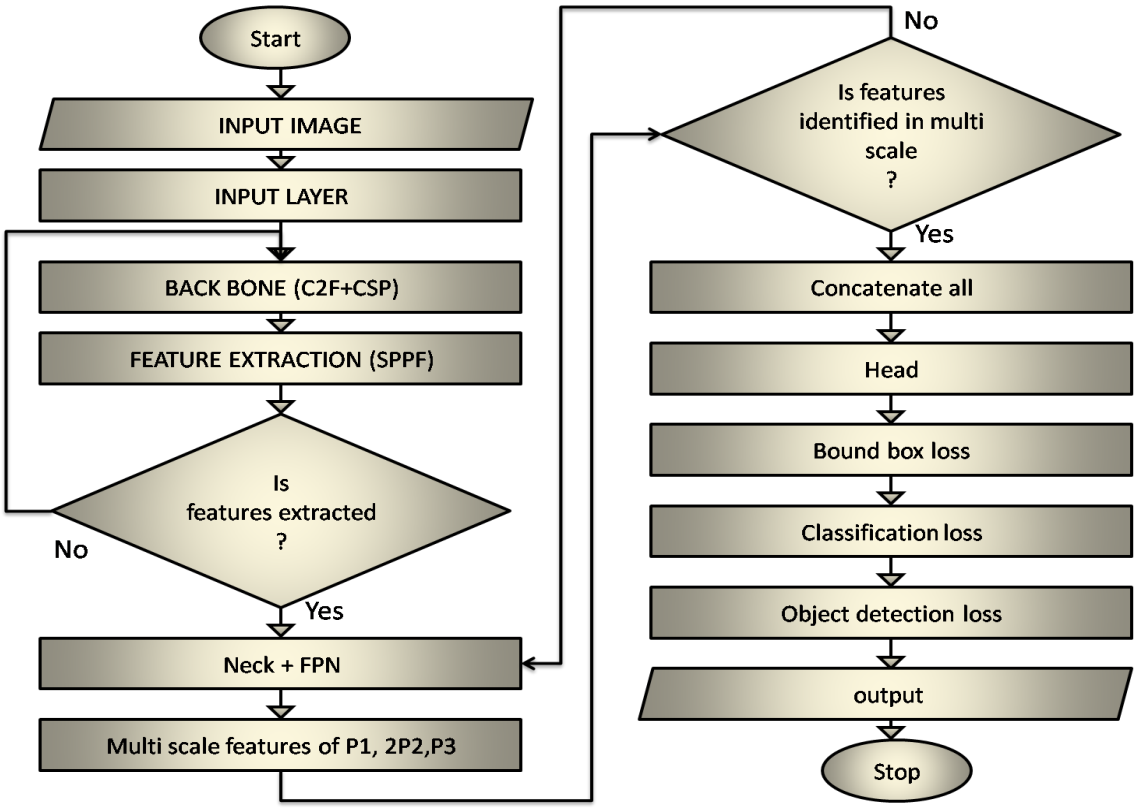


Fig-4: Flowchart of the Benchmark YOLOV8N

The input representation of the images $I \in \mathbb{R}^H * W * 3$ is resized to fixed size of 6406403. The normalized image is presented as Equation 1;

$$I' = \frac{I}{255} \quad (1)$$

Where I is the image in the normalized input data of I' in equation 1. The back bone of the YOLOV which combined Cross stage partial (CSP) connection fast (C2F) layer to reduce the computation while maintaining the reuse of features. With N layer, the C2f is presented as;

$$x_0 = Conv\ (x) \quad (2)$$

$$x_1 = Conv\ (x) \quad (3)$$

$$x_2 = Conv\ (x) \ldots . x_n = Conv\ (x_{n-1}) \quad (4)$$

$$y = Concat\ (x_1, x_2, \ldots \ldots . x_n) \quad (5)$$

$$z = Conv\ (y) \quad (6)$$

The idea is to reduce repeated feature extraction by reusing and merging earlier feature maps. The model in the neck uses feature pyramid network and path aggregate network combined for multi scale feature identification. Let the feature maps at different scales be represented as P_3, P_4, P_5. The FPN is presented as Equation 7;

$$P_4' = Upsample\ (P_5) + P_4 \quad (7)$$

$$P_5' = Upsample\ (P_4') + P_3 \quad (8)$$

Neck

$$P_4'' = Downsample\ (P_3') + P_4' \quad (9)$$

$$P_5'' = Downsample\ (P_4') + P_5 \quad (10)$$

This ensures both high-level semantic and low-level spatial information is retained. The head of the model used anchor free head output in equation 11;

$$Output = [x, y, w, h, c, p_1, p_2, \dots\dots, p_N] \quad (11)$$

Where (x, y) is the object centre, (w, h) are object width and height, c is the confidence score and p is the probability of object belonging to class i. For the output location, the class score is presented in equation 12.

$$Class\ score = \sigma(c).Softmax([p_1 \dots\dots, p_N]) \quad (12)$$

The bounding box loss is Equation 13, object detection loss is Equation 14, classification loss is equation 3.15, and then the overlapping bounding box loss is Equation 16.

$$L_{cIoU} = 1 - CLoU(b, b^{gt}) \quad (13)$$

$$L_{obj} = BCE(c, c^{gt}) \quad (14)$$

$$L_{cls} = \sum_{i=1}^{N} BCE(p_i, p_i^{gt}) \quad (15)$$

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \quad (16)$$

Boxes with IoU > threshold (0.5) and lower confidence are suppressed. The Table 1 presents the equation description.

Table 1: Model description of the yolov-8n

Equation	Description in this work
3.1	Normalized input image
3.2	Convolutional layer
3.3	1st convolutional layer
3.4	2nd convolutional layer
3.5	Combined features from layer 1 and 2
3.6	Output features
3.7	Features of scale P4
3.8	Features of scale P5
3.9	Down-sampling features of P4
3.10	Down-sampling features of P5
3.11	Output features to the head
3.12	Classification score
3.13	Bounding box loss
3.14	Object detection loss
3.15	Classification loss
3.16	IOU or suppression of overlapping bounding boxes

C. The Anti-Occlusion Management Model

First the visibility of the YOLOV-8N was enhanced as shown with the object in the image defined as O, the object centre (x, y) the width and height (w, h) and the class label C. where the visibility score is defined as $v_O = 1: visible;\ and\ v_o = 0: occlusion$. Where O is defined as Equation 17 (He et al., 2016);

$$O = \{x, y, w, h, C\} \quad (17)$$

Where M_o *be the binary mask of the object and* MQ_o *be the visible part of occlusion.*

$$v_O = \frac{Area(MQ_o)}{Area(M_o)} = \frac{|MQ_o|}{|M_o|} \quad (18)$$

The Equation 18 was applied to predict the visibility score of detecting occluded object. To penalize missed detection, we introduce the visibility aware weighted loss function as;

$$L_{occl} = \sum_{i=1}^{N}(1 - \gamma v_i)\ .\ L_{det}^{i} \quad (19)$$

If $v_i \approx 1$: Object is clear, normal loss; If $v_i \approx 0$: object is occluded

Context aware feature extraction improvement was also added to optimize feature extraction quality in the backbone. The feature maps from the backbone are defined as $F \in \mathbb{R}^H * W * C$. The contextual attention module A: $\mathbb{R}^H * W * C \rightarrow \mathbb{R}^H * W * 1$ such that:

$$\alpha = A(F) = \alpha(W_2 . ReLU\ (W_1 . F)) \quad (20)$$

Then the improved features are given as;

$$F' = F \odot \alpha \quad (21)$$

Where $\odot$ is the element wise multiplication, α is the attention mask which gives clue of occlusion region.

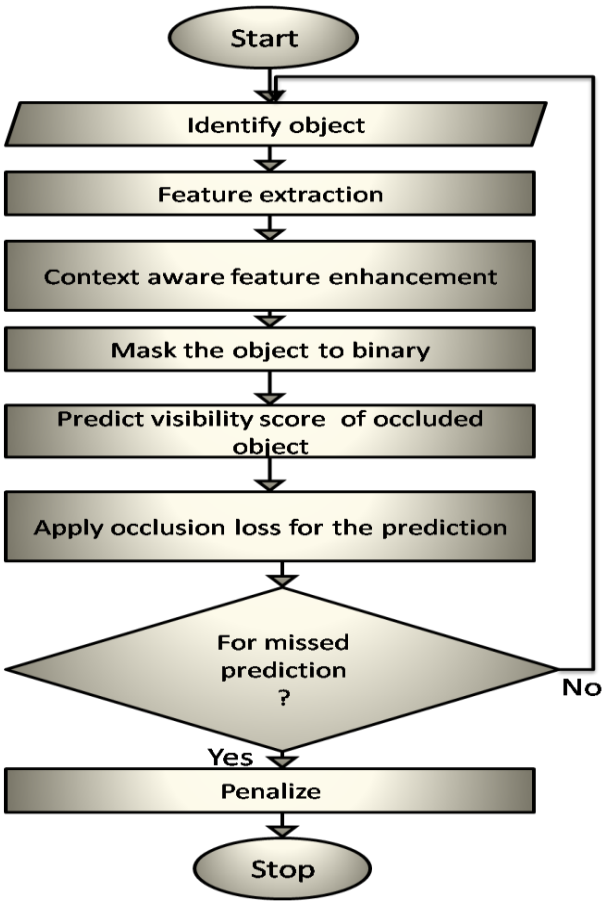


Fig-5: Flow Chart of the Occlusion Detection Algorithm

The flowchart in Figure 5 outlines the step-by-step process of an anti-occlusion object detection model, designed to detect objects even when they are partially hidden. It starts with the "Identify object" phase, where the model attempts to detect any object present in the image, whether it's partially visible or not. Once an object is detected, the "Feature extraction" stage follows, where the model gathers important visual details from the image using convolutional layers these features help recognize patterns like edges, textures, and shapes. Then, "Context-aware feature enhancement" takes place. This step boosts the model's understanding of the surrounding area of the object, helping it infers what might be hidden by learning from nearby visible elements.

After context enhancement, the system proceeds to "Mask the object to binary", where the detected object is converted into a black-and-white mask (1 for visible parts and 0 for occluded or background), making it easier to compute visibility. The "Predict visibility score of occluded object" phase quantifies how much of the object is actually visible in the image this score influences how the model treats the object during learning. Next, in "Apply occlusion loss for the prediction", a specialized loss function penalizes the model more for errors involving occluded objects, encouraging it to learn from challenging cases. The decision block "For missed prediction?" checks if the model failed

to detect an object it should have. If the answer is Yes, it proceeds to "Penalize" increasing the training error to make the model try harder next time. If the answer is No, it skips penalization. Finally, the process ends at the "Stop" node. Each step works together to help the model become better at recognizing partially visible objects in complex real-world scenes.

(a) Distance Estimation Algorithm (DEA)

The DEA algorithm constitutes key camera information such as the bounding box size, pixel distance, coordinate of the bounding box, to calculate the distance between camera and object. Let the camera focal length be defined as F, the principal points of coordinate for image captured defines as P_x and P_y. The size of object defined as (x_1, y_1) for the top left and (x_2, y_2) for the bottom right corner of the detected image bounding box. Object weight defined as $W_{image} = (x_2 - x_1)$, object height defined as $H_{image} = (y_2 - y_1)$ is the height of the bounding box, where D_{real} define the actual object weight and height. The camera distance from the image is measured using Equation 22

$$D = \frac{F * D_{real}}{W_{image} * H_{image}} \quad (22)$$

Where W_{image} is the weight of the image bounding box, while F is given as Equation 23;

$$F = \frac{Object\ size * bounding\ box\ size}{object\ size}. \quad (23)$$

The Equation 23 is used to measure the distance of object. From the results obtained, we assume a fixed distance of 20meters as the threshold to classify obstacle. The measurement from DEA is compared with the set value and upon ≤ 20m, the object with the least distance is classified as obstacle.

(b) Text to speech model

The text to speech model was tailored towards converting the output of the deep learning model + distance estimation output to speech and informs the user of obstacle. The model is made of our main components which are the processor, encoder, decoder and vocoder as shown in Figure 6. The input from the deep learning models is preprocessed into phonemes which are the smallest unit of sound. This is then transformed into numeric embedding by the processor and passed to the encoder for analysis into contextual and structural patterns. The encoded data are feed to the decoder which then generates the acoustic version of the features as mel-spectrogram, representing the sequences of rhythm, pitch and speech converted by the vocoder to sound.

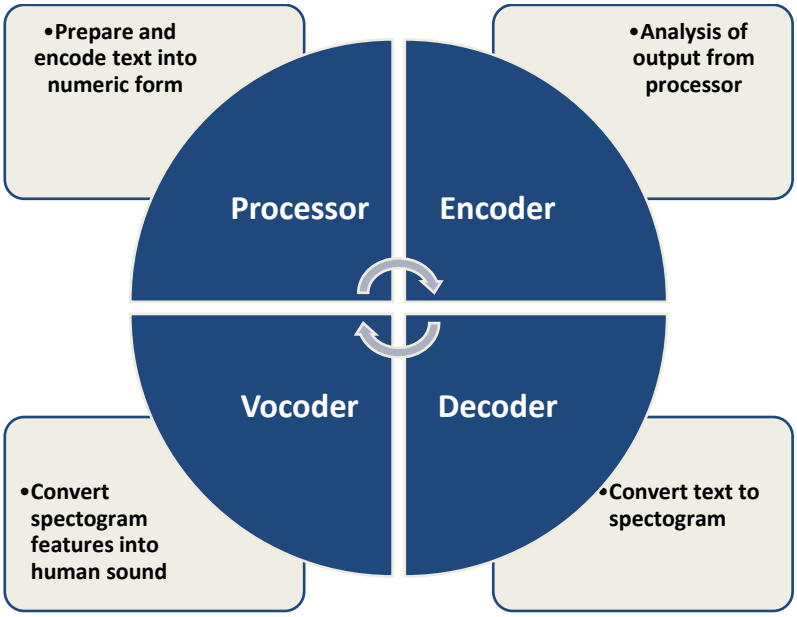


Fig-6: Text to speech model (Yu et al., 2023)

(c) Kalman filter model

The Kalman Filter is a powerful mathematical tool used to estimate the state of a dynamic system in the presence of noise and uncertainty. It is particularly useful for tracking the behaviour of moving objects because it continuously

predicts and updates an object's position and velocity over time, even if observations are noisy or partially missing. The object state space is defined as Equation 24;

$$X_t = \begin{bmatrix} x_t \\ y_t \\ \hat{x}_t \\ \hat{y}_t \end{bmatrix} \quad (24)$$

Where $\hat{x}_t$ and $\hat{y}_t$ are the components of velocity, while x_t and y_t are position components. This state is tracked as the object moves. The transition state of the object is defined as Equation 25;

$$x_{t|t-1} = F.x_{t-1|t-1} + u_t + w_t \quad (25)$$

Where F is the state transition which is the motion model, u_t is the control input, and w_t is the process noise. To predict the uncertainty in object the Equation 26 was applied;

$$P_{t|t-1} = F.P_{t-1|t-1}.F^T + Q \quad (26)$$

Where P is the covariance matrix and Q is the processing noise covariance. The data captured from camera is presented as Equation 27;

$$z_t = \begin{matrix} x_t^{meas} \\ y_t^{meas} \end{matrix} = H.x_t + v_t \quad (27)$$

Where H is the observation matrix and vt is the noise in the data. The Kalman gain is presented as Equation 28, update state estimate with Equation 29 and then update covariance with Equation 30;

$$K_t = P_{t|t-1}.H^t.\left(H.P_{t|t-1}.H^t + R\right)^{-1} \quad (28)$$

$$x_{t|t} = x_{t|t-1} + K_t.(z_t - H.x_{t|t-1}) \quad (29)$$

$$P_{t|t} = (I - K_t.H).P_{t|t-1} \quad (30)$$

(d) Assistive Navigation Guide System

This section presents the integrated system which facilitates autonomous navigation for the blind. The system is made of several components which are the object detection model, designed with anti-occlusion algorithm, Kalman filter for dynamic object tracking, distance estimation measures distance of objects to address same object occlusion problem, then the classified object is filtered and transcribed to speech through the text to speech model. This speech notifies the user of the object detected as obstacles. Figure 7 presents the system flow diagram.

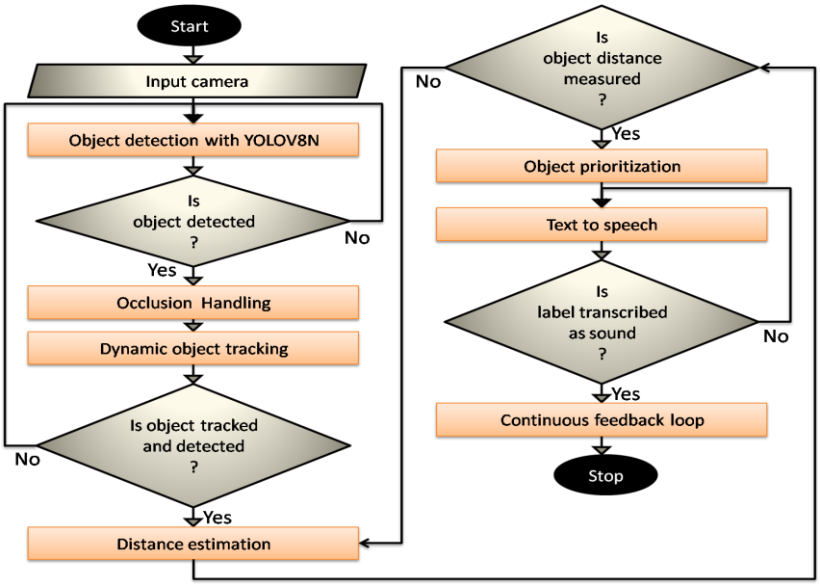


Fig-7: Flow chart of the Assistive Navigation Guide System

D. System Implementation

The implementation of the intelligent assistive system for visually impaired users was modular and sequential, ensuring seamless integration of all functional components. The object detection module was implemented using YOLOv8n, a lightweight real-time convolutional neural network model. This model was trained on a custom dataset containing everyday objects relevant to urban navigation, such as people, vehicles, doors, and chairs. To enhance the robustness of object detection, especially in obstructed scenes, an anti-occlusion model was incorporated. This model was developed using attention mechanisms and contextual refinement layers to ensure that partially visible objects were still recognized and processed accurately.

For tracking the behaviour of dynamically moving objects across video frames, the Kalman Filter algorithm was implemented. It estimated the current position and predicted the next state of each detected object by leveraging both the observed data and previous predictions, helping maintain object identity and suppress noise. Following detection and tracking, a distance estimation module was implemented using a monocular depth approximation model, which calculated the approximate distance between the user and each object based on bounding box size and camera calibration parameters. This allowed the system to prioritize objects within the user's immediate path.

Finally, a Text-to-Speech (TTS) engine was implemented using a Tacotron-based encoder-decoder architecture combined with a vocoder for natural voice output. The system generated structured sentences describing the object type, its relative location, and estimated distance (e.g., "There is a person in front of you"). These messages were fed to the vocoder to produce real-time audio guidance, which was output through a earphone for the visually impaired user. All modules were developed and integrated using Python, OpenCV, PyTorch, and relevant deep learning libraries, and were tested on a Jetson Nano platform with a connected camera for live deployment.

3. Results Of the Deep Learning Model Training

This training of the YOLOV-8N was evaluated considering the metrics such as precision, recall, mean Average Precision (mAP) and Absolute Precision. The results obtained are reported Table 2.

Table 2: Training and validation results

Metrics	Training	Validation
Focus loss	0.2141	0.2505
Object detection loss	0.2303	0.2553
Classification loss	0.2477	0.2702

Table 2 is grouped into training and validation considering bounding box loss, object detection loss and classification loss. The loss results measured the error in correctly classifying object, bounding box and label classification label. The results showed that at the model performance convergence with tolerable loss value which approximates zero for each loss. For instance, the focus loss recorded 0.21 for training and then 0.25 for validation, object detection loss recorded 0.23 for training and 0.26 for validation, then classification loss recorded 0.25 for training loss and 0.27 for validation loss. In Table 3, the recall, precision, and mAP were also measured and reported. All these values were achieved automatically in the training tool using the relationship between the test, training and validation images in the dataset.

Table 3: Overall model performance

Metrics	Results
Precision	0.974
Recall	0.952
mAP@0.5	0.869
Map@0.5-0.9	0.571

The precision which measured the probability of correctly classifying objects reported 0.97, recall reported 0.95, mean absolute precision (mAP) at 0.5 IoU reported 0.86 and mAP at 0.5-0.95 IoU repoted 0.57. The precision revealed that the model ability to correctly classify images of obstacle is at 97% success rate, the recall implied that the classification success of actual obstacle correctly is 95%, the mAP also implied that the average classification success

of the model is 86% and 57% success rate in occluded environment. Table 4 compared the results with others in literature.

Table 4: Comparative Analysis with Other Obstacle Detection Models

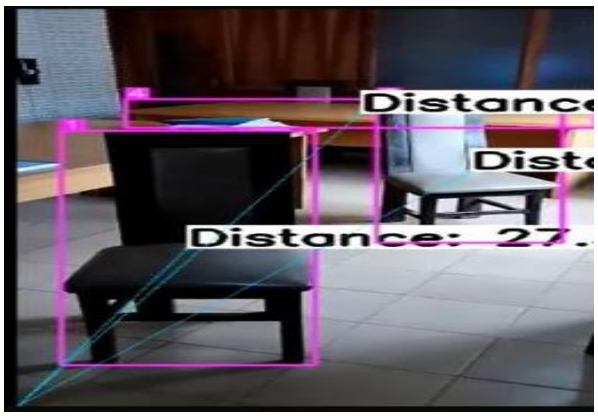
Author	Models	Precision	Recall	mAP
Xu et al. (2021)	SSD	72.8	66.5	70.1
Cao et al. (2023)	Faster RCNN	72.8	66.5	76.4
Chen et al. (2023)	YOLOv5n	86.7	86.8	88.8
Jiang et al. (2022)	TPH-YOLO-V5	92.7	82.3	88.4
Nazir et al. (2023)	YOLO-V7tiny	73.5	81.2	81.8
Rath et al. (2023)	YOLO-V8n	90.3	80.7	86.3
Salam et al. (2021)	Golf-YOLO	83	95	87.9
Our work	YOLOV-8n + Anti occlusion + DEA	97.4	95.2	86.9

Table 4 compared the results obtained from this work with other related work in literature. From the results, it was observed that our model competes among the best. The reason was because of the introduction of the occlusion aware algorithm which improved the feature extraction quality and also suppression of overlapping bounding boxes.

A. Experimental Testing of the system

This section presents the experimental testing on the model for obstacle avoidance. The test data were used as the input data to test the model. The results obtained are grouped into indoor result and outdoor result. Figure 8 presents the result of the indoor test considering chairs as the object of interest.

Case (A) test result for indoor



CASE (B) test result at Indoor office

Fig-8: Practical Validation Results for Indoor

The Figure 8 presents the result of the indoor test carried out with the developed model with chairs as obstacle. From the results, it was observed that the object of interest captured by the camera was correctly classified with their respective distance measured to identify obstacles. The results showed that the object were correctly detected and classified in the image, while notifying the user of the obstacle through sound. Figure 9-11 presents the outdoor result for different object of interest.

Fig-9: Tricycle test in outdoor

Fig-10: Person test in outdoor

Fig-11: Person and door test result

The results of Figure 9 to 10 present the different outcome when classifying obstacles at different outdoor positions. The obstacles considered are person, door, and tri-cycle. The results showed that all the object of interest were correctly classified and their position measured correctly to detect obstacle and then notify user of the problem.

4. Conclusion

In this study, the design and implementation of an intelligent assistive navigation guide system for blind and visually impaired people with the aid of deep learning techniques is presented. This paper was inspired by the growing incidence of vision impairment around the world and the shortcomings of current assistive technologies for supporting accurate, affordable, and reliable navigation assistance. To address these challenges, a smart navigation framework based on YOLOv8 for real-time obstacle detection and an anti-occlusion management model to recognize objects that are partially covered were designed. Other modules such as DEA, Kalman filter-based object tracking, and a Text-to-Speech system were added to enable the identification of obstacles, measurement of distance, prediction of motion and voice guidance to users in real-time.

The system was trained and validated with a unified dataset of primary image data collected with the ZED 2i camera and secondary images in the Microsoft COCO dataset. The experimental results showed that the models have good performance in both indoor and outdoor environments. The proposed model had a precision of 97.4%, a recall of 95.2% and an mAP@0.5 of 86.9% demonstrating the ability to detect and classify navigation obstacles like chairs, door, person, car and tricycle. The comparative study with related models revealed that the presented model was competitive and robust, where the anti-occlusion feature enhancement and visibility-aware loss function were effective in enhancing the robustness of the model, especially in the occluded environment.

Finally, it was demonstrated that the navigation guide system developed in this study is effective as a real-time navigation assistance system for blind people, by accurately detecting the surrounding obstacles, estimating the distance to them, and giving immediate navigation guidance information to blind people in the form of audio feedback for safe navigation. Deep learning combined with speech interaction between the user and the computer provides a practical and intelligent solution to make the population of people with visual impairments more independent, and thereby improve their mobility. The research paves the way for further advancements in AI-driven assistive technologies and showcases the capabilities of intelligent vision systems for improving accessibility and quality of life. Future work might extend the database to contain more obstacle types, enhance the performance in extreme lighting and weather conditions, as well as optimise deployment on wearable low-power devices to achieve wider real-world adoption.

5. References

[1] Anna, F. (2024). World Braille Day; vision loss predicted to surge by 55% in 2050. *Statista*. https://www.statista.com/chart/31502/expected-number-of-people-with-vision-loss-globally/

[2] Atitallah, A., Said, Y., Atitallah, A., Albekairi, M., Kaaniche, K., & Boubaker, S. (2024). An effective obstacle detection system using deep learning advantages to aid blind and visually impaired navigation. *Ain Shams Engineering Journal, 15*, 102387. https://doi.org/10.1016/j.asej.2023.102387

[3] Burhan, A. M., Klahan, B., Cummins, W., Andrés-Guerrero, V., Byrne, M. E., O'Reilly, N. J., Chauhan, A., Fitzhenry, L., & Hughes, H. (2021). Posterior segment ophthalmic drug delivery: Role of muco-adhesion with a special focus on chitosan. *Pharmaceutics, 13*, 1685. https://doi.org/10.3390/pharmaceutics130101685

[4] Cao, F., Xing, B., Luo, J., Li, D., Qian, Y., Zhang, C., Bai, H., & Zhang, H. (2023). An efficient object detection algorithm based on improved YOLOv5 for high-spatial-resolution remote sensing images. *Remote Sensing, 15*, 3755. https://doi.org/10.3390/rs15153755

[5] Chen, H., Chen, Z., & Yu, H. (2023). Enhanced YOLOv5: An efficient road object detection method. *Sensors, 23*, 8355. https://doi.org/10.3390/s23208355

[6] Ebere, C. E. U., Udanor, C. N., & Anoliefo, E. (2024). Exploring the depths of visual understanding: A comprehensive review on real-time object detection techniques. *Preprints*. https://doi.org/10.20944/preprints202402.0583.v1

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In *European Conference on Computer Vision* (pp. 630–645). Springer.

[8] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of YOLO algorithm developments. *Procedia Computer Science, 199*, 1066–1073. https://doi.org/10.1016/j.procs.2022.01.135

[9] Oluwaseyi, E., Martins, E., & Abraham, E. (2023). A comparative analysis of YOLOv5 and YOLOv7 object detection algorithms. *Journal of Computing and Social Informatics, 2*(1), 1–12.

[10] WHO. (2023). Blindness and visual impairment. *World Health Organization*. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[11] Yu, J., Xu, Z., He, X., Wang, J., Liu, B., Feng, R., Zhu, S., Wang, W., & Li, J. (2023). DIA-TTS: Deep-inherited attention-based text-to-speech synthesizer. *Entropy, 25*(1), 41. https://doi.org/10.3390/e25010041