Indonesian Journal of Electrical Engineering and Computer Science
Vol. 43, No. 2, August 2026, pp. 555~567
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v43.i1.pp555-567

Low-cost real-time campus assistive navigation device for the visually impaired: The University of Ilorin case study

Mahmud Hafeez Owolabi[1], Idajili John Ojochegbe[1], Ayinla Shehu Lukman[1], Jimoh-Mahmud Aishat Oladayo[2], Yusuf Abdulrahman Olalekan[1], Olaogun Jerry Oluwajomiloju[1]
[1]Department of Computer Engineering, Faculty of Engineering and Technology, University of Ilorin, Ilorin, Nigeria
[2]Department of Telecommunication Science, Faculty of Communication and Information Sciences, University of Ilorin, Ilorin, Nigeria

Article Info

Article history:

Received Mar 3, 2026
Revised Jun 19, 2026
Accepted Jun 27, 2026

Keywords:

Campus assistive navigation
Computer vision
GPS
Offline navigation
Raspberry Pi 4

ABSTRACT

Navigating dynamic environments remains a significant challenge for visually impaired individuals due to limited spatial awareness, which restricts their mobility, independence, and safety. Traditional aids such as white canes and guide dogs provide physical support but lack contextual feedback. While recent advances in artificial intelligence (AI) and computer vision (CV) have enabled real-time sensing, many assistive systems remain costly, complex, and dependent on continuous network connectivity. This study introduces a low-cost, portable campus assistive navigation (CAN) device that delivers real-time perception and offline guidance. The system employs a custom dataset, an optimized YOLOv11 model, and a Pi Camera for continuous object detection, supported by an HC-SR04 ultrasonic sensor for obstacle avoidance and auditory alerts. Training and validation confirmed robust convergence, with precision ~0.95, recall near 1.0, and mAP@0.5 exceeding 0.9, while mAP@0.5:0.95 remained above 0.8, demonstrating reliable detection and generalization under strict thresholds. Field tests further reported confidence scores of 0.74–0.84, 98% accuracy in distance measurement, and GPS localization within ±1.5 m. Real-time auditory and haptic feedback via Bluetooth headphones enhanced mobility and safety. The CAN device offers a scalable, affordable solution for autonomous navigation in campus environments.

Corresponding Author:

Mahmud Hafeez Owolabi
Department of Computer Engineering, Faculty of Engineering and Technology, University of Ilorin
P.M.B. 1515, Ilorin, Nigeria
Email: mahmud.ho@unilorin.edu.ng

1. INTRODUCTION

Visual impairment refers to a partial or complete loss of visual function that limits the ability to perceive and interpret the surrounding environment, commonly caused by uncorrected refractive errors, cataracts, and glaucoma [1], [2]. It remains a major global health concern with profound social and economic implications. Current statistics indicate that more than 2 billion people worldwide are affected, and this number is projected to exceed 3.5 billion by 2050 [3]. The resulting reduction in environmental awareness makes navigation difficult and often restricts access to education, employment, and social participation, particularly in resource-limited regions [4]. These challenges highlight the need for affordable, intelligent assistive technologies that support independent mobility among visually impaired persons.

Conventional mobility aids, such as white canes and guide dogs, remain essential in daily navigation but are constrained in functionality. White canes offer tactile guidance only within a short detection range of approximately 1–2 m and cannot identify above-ground or rapidly moving hazards. Guide dogs assist with path following and obstacle avoidance, but do not communicate detailed spatial information, object context, or environmental changes to the user. Their performance may also vary with training quality, fatigue, and environmental distractions, while the cost and maintenance requirements render them inaccessible to many individuals. Consequently, both tools provide reactive, rather than predictive, awareness and rely heavily on prior familiarity with the environment. These limitations have accelerated the development of technology-enabled assistive solutions that integrate artificial intelligence (AI), computer vision (CV), and embedded sensing to enhance navigation autonomy [5]. Real-time perception systems can extend awareness beyond physical contact, offering semantic understanding of surroundings [6]. However, many existing devices remain expensive, computationally intensive, and dependent on continuous internet connectivity, which limits their portability and usability across diverse environments.

Several AI-driven assistive technologies for people with disabilities and visual impairments have evolved through research focused on enhancing mobility [7], [8] and on improving social and formal engagement [9], [10]. Early systems, such as [11], [12], used tactile graphics and low-light enhancement but relied on client–server setups, which limited portability. Introduced a local obstacle detection model, which still required internet access for key functions, reducing its utility in low-bandwidth environments. A study employed single shot multibox detector (SSD)-based object detection with distance estimation, though low-light performance remained a challenge [13].

You only look once (YOLO) is a single-pass detection model [14] that improves efficiency over traditional scanning. This was leveraged in some wearable devices with sensors, voice interaction, and augmented reality (AR), but its dependence on constant connectivity hindered remote use. Studies [15] and [16] developed offline YOLOv3-based systems with distance sensors for navigation and text recognition; however, the minimal audio feedback limited situational awareness. [17] used YOLOv5 in smart glasses for object detection, but lacked navigation support. In an advanced work that combined CV and voice guidance, reliance on generic datasets reduced accuracy in identifying key features such as crossings and doorways [18].

Research into affordable assistive devices has explored smart glasses with optical character recognition (OCR), GPS, facial recognition, and ultrasonic sensors [19]-[21]. However, their complexity raised concerns about power usage without improved navigation. Innovations like smart glasses and shoes added ground-level hazard detection, but a limited range reduced their effectiveness in reducing reaction time [22]. A study used YOLOv3 with voice alerts for obstacle detection [23]. However, it struggled to adapt across indoor and outdoor settings with inconsistent recognition. In advanced studies, the field was enhanced with YOLO-insight, which integrates a YOLO model, OCR, ultrasonic sensors, multilingual text-to-speech (TTS), and natural language processing (NLP) [24], [25]. However, the implementation is expensive due to high hardware requirements, and reliance on generic datasets impairs the recognition of key navigational elements.

Collectively, the reviewed studies reveal a gap: the need for a portable, low-cost system that offers accurate, context-aware detection and navigation without internet dependence. To address these challenges, this study proposes a low-cost, portable campus assistive navigation (CAN) system: a compact, Raspberry Pi 4–based wearable that integrates a Pi camera, ultrasonic sensors, and a NEO-6M GPS module for vision, proximity sensing, and location tracking. A custom YOLO11-n model, trained on navigation-critical objects such as crossings, doorways, and staircases, enables efficient, low-power inference directly on the device. Offline OpenStreetMap data ensures reliable indoor and outdoor navigation across the University of Ilorin campus, while audio guidance and haptic feedback via Bluetooth AirPods enhance situational awareness. Supported by energy-efficient components and a 20,000 mAh power bank for extended offline use, the CAN system combines affordability, portability, and real-time perceptual intelligence, advancing inclusive mobility solutions in alignment with sustainable development goals (SDGs) 10 and 11. The contributions of this study are: i) low-cost, portable navigation system: Introduction of a CAN device and a Raspberry Pi 4 wearable that integrates a camera, ultrasonic sensors, and GPS for vision, proximity, and location tracking; ii) offline geospatial integration: utilization of OpenStreetMap data to enable reliable indoor and outdoor navigation without continuous internet connectivity; iii) multimodal feedback for awareness: provision of audio guidance and haptic cues via Bluetooth AirPods to enhance user safety and interaction; iv) energy-efficient extended operation: incorporation of low-power components and a 20,000 mAh power bank to support prolonged offline use in real-world contexts.

2. METHOD

The CAN system framework integrates hardware and software into a unified structure comprising three key functions: object recognition, obstacle detection with distance measurement, and route navigation, as illustrated in Figure 1. A Raspberry Pi 4 acts as the central processor, coordinating inputs from a Pi camera, ultrasonic sensors, and the GPS module. The camera streams video for AI-based object recognition, while the ultrasonic sensors detect hazards beyond the camera’s view and measure distances up to 3 m.

GPS supports navigation. Processed information is delivered to the user through TTS and haptic feedback, enabling real-time situational awareness and ensuring safe mobility.

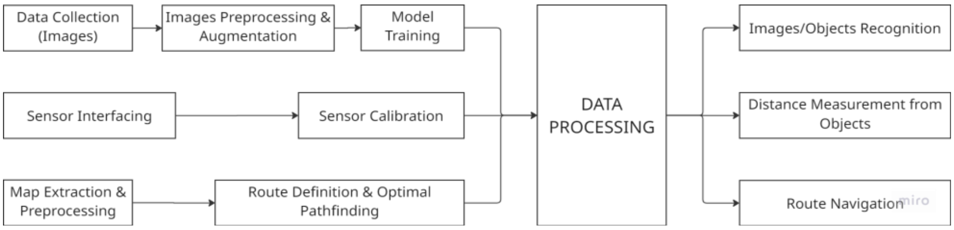

Figure 1. CAN framework

2.1. Data collection, preprocessing, and augmentation

The object recognition system began with building a custom dataset tailored to the University of Ilorin campus. Images were captured at various locations, focusing on key obstacles that pose significant navigational challenges for visually impaired individuals, such as cars, pedestrians, potholes, staircases, and entrances. To increase diversity, contextually relevant images from the internet were added. A total of 896 images were gathered, spanning 15 object categories, including a stop sign, person, bicycle, bus, truck, car, motorbike, reflective cone, ashcan, warning column, spherical roadblock, pole, dog, tricycle, and fire hydrant. The dataset was uploaded to Roboflow for preprocessing, as illustrated in Figure 2. This stage involved cleaning, annotating, augmenting, and resizing images to ensure compatibility with YOLO11-n. The dataset was partitioned into training, validation, and test subsets in an 80:10:10 split. Augmentation techniques such as random rotations, flipping, scaling, brightness adjustments, and noise injection expanded the training dataset from 716 to 2,398 samples, increasing feature diversity and reducing sensitivity to variations in illumination, pose, and scale.

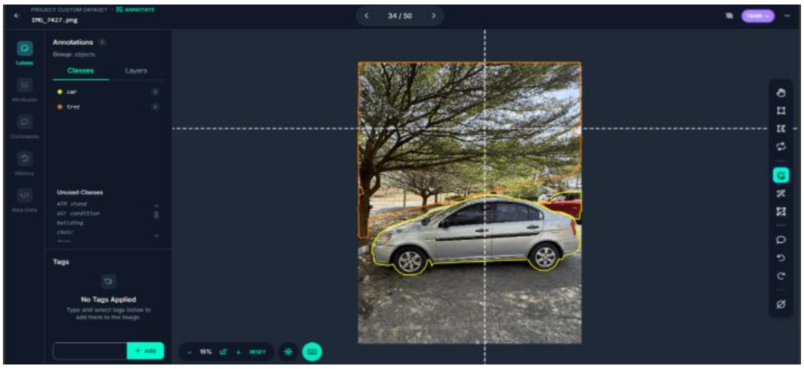

Figure 2. Bounding box annotation for object labeling

2.2. Model training, fine-tuning, and output classification

The YOLO11-n was adopted as the detection model, providing superior accuracy, speed, and efficiency compared to prior YOLO architectures [26]. The convolutional neural network (CNN) is a general deep learning architecture used for image preprocessing, feature detection, segmentation, 3D reconstruction, object recognition, and motion analysis [27], the model is trained on raw data, capitalizing on its ability to process time series and autonomously extract complex patterns through convolutional and pooling layers [9], [10], whereas YOLO is a CV model with a specialized object detection algorithm built on CNN principles. The model was fine-tuned on the custom object-detection dataset following transfer learning. Training was performed using the Ultralytics framework, which integrates an optimized pipeline with improved augmentations and more stable convergence behavior on YOLOv11 than earlier YOLO releases. The model was trained at a resolution of 640×640 for 100 epochs, with a batch size of 32 and a learning rate of 0.001. Performance was tracked via loss curves, precision, recall, and mean average precision (mAP) metrics, all of which showed consistent improvements in detection accuracy. After training, the model was exported from PyTorch to the open neural network exchange (ONNX) format, then converted to the novel convolutional neural network (NCNN) format for optimized deployment on embedded hardware, such as the Raspberry Pi, enabling faster inference and reduced memory usage.

The fine-tuned model was deployed on the edge device and integrated in a Python script (object_detection.py) to process live video from the Pi camera. It generated bounding boxes and labels per frame, with usability enhancements: detections below 50% confidence were discarded, as illustrated in Table 1; a cap of three objects per frame minimized cognitive load; and a TTS module delivered real-time audio description.

Table 1. Factors affecting the choice of confidence threshold

Confidence range (%)	Objects detected per second	Confidence level	Key risk
0–29	≤15	Low	Misleading feedback, safety compromised
30–49	≤10	Low	User misled due to poor reliability
50–59	≤5	High	Accurate, balanced feedback; avoids overload
60–79	1–2	Very high	Missed obstacles → collision risk
80–100	≤1	Extremely high	Delayed scanning → collision risk

2.3. Distance measurement (obstacle avoidance) module

The HC-SR04 sensor operates by transmitting an ultrasonic pulse and measuring the time interval between the transmission and the echo's return to determine distance as expressed in (1). During calibration, raw distances in centimeters measured by the sensor, as given in (2), were compared with actual distances measured with a measuring tape.

$$Distance = \frac{V \times time}{2} \tag{1}$$

Here, v is the speed of sound in air (≈ 343 m/s). In practice, this is implemented as in (2).

$$distance\ (cm) = pulse\ duration\ x\ 17150 \tag{2}$$

Using the equation of a straight line as expressed in (3), where Y is the uncalibrated sensor output distance, X is the real distance using a measuring tape, $\hat{a}$ is the slope defined in (4), and $\hat{b}$ is the intercept as defined in (5).

$$Y = \hat{a}X + \hat{b} \tag{3}$$

$$\hat{a} = \frac{\sum X \sum Y - n \sum XY}{(\sum X)^2 - n \sum X^2} = 0.957977 \tag{4}$$

$$\hat{b} = \frac{\sum X \sum XY - \sum Y \sum X^2}{(\sum X)^2 - n \sum X^2} = -0.82879 \tag{5}$$

The calibrated sensor output, $\bar{Y}$ is thus expressed as in (6) from (3) and rearranged in (7) by substituting (2), (4), and (5).

$$\bar{Y} = \frac{\mathrm{Y} - \hat{b}}{\hat{a}} \tag{6}$$

$$\bar{Y} = (17902.309 \times pulse\ duration) + 0.86514423 \tag{7}$$

In (7), the distance sensor output is derived from (2)-(6) and implemented in the hardware integrated development environment. The distance is measured in cm. To minimize user cognitive load, sensor data were preprocessed: objects beyond 300 cm were ignored, as they pose no immediate risk, while those between 100–300 cm triggered speeches distance alerts ("An object is 1 m and 35 cm in front of you"). For objects closer than 100 cm but more than 50 cm, the system activated a 5 Hz proximity-based beeping, while the frequency increased to 10 Hz with the object's distance less than 50 cm. The distance alert (A_d) is mathematically expressed in (8).

$$\mathrm{A}_d = \left\{ \begin{array}{c} 0 \\ sound(speech) \\ sound(5\ Hz\ beep) \\ sound(10\ Hz\ beep) \end{array} \middle| \begin{array}{l} for: \qquad d > 300\ cm \\ for: 100\ cm < d \le 300\ cm \\ for: \ 50\ cm < d \le 100\ cm \\ for: \qquad d \le 50\ cm \end{array} \right\} \tag{8}$$

2.4. Offline map extraction and processing

To support offline use, the navigation system utilizes a streamlined OpenStreetMap (OSM) dataset focused solely on the University of Ilorin campus, as shown in Figure 3. The dataset was sourced from and refined using the OSM export tool, minimizing file size and optimizing performance. A database of GPS coordinates for key campus landmarks, such as administrative offices, lecture halls, the library, hostels, and religious and recreational centers, using field data from a NEO-6M GPS module and validated against OSM to ensure accuracy, especially for the visually impaired. The navigation graph is built on geographic coordinates, with landmarks as nodes and connecting paths (roads and walkways) as edges. Each edge is weighted by physical distance between nodes, computed using the Haversine formula given in (9) to determine great-circle distances on the Earth.

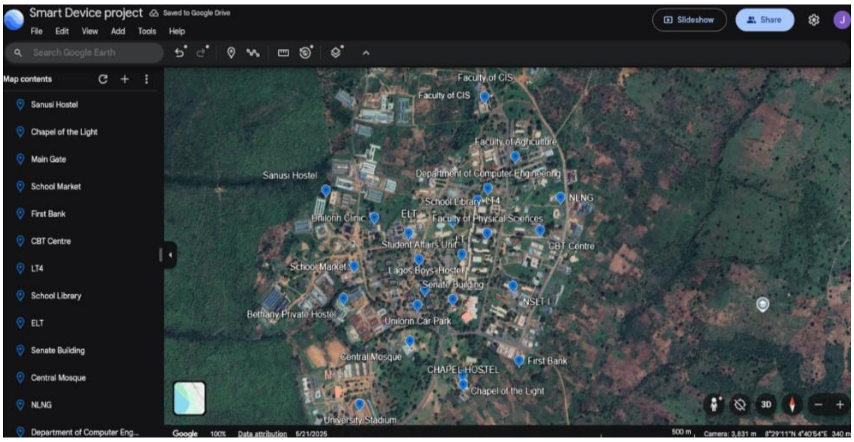

Figure 3. University of Ilorin offline map with major landmarks

$$d = 2R\ \sin^{-1}\left(\sqrt{sin^2\left(\frac{\Delta\emptyset}{2}\right)}\ +\ \cos\emptyset_1\ \cos\emptyset_2\ sin^2\left(\frac{\Delta\lambda}{2}\right)\right) \tag{9}$$

The Haversine formula in (9) calculates the distance d between two points using Earth's radius (R ≈ 6371 km), latitude (∅), and longitude (λ). This distance-weighted graph enables Dijkstra's algorithm to compute optimal routes from the user's location to a target destination. During navigation, real-time GPS coordinates are matched with reference nodes to generate step-by-step instructions, such as "Proceed 30 m forward, then turn north towards the library." Figure 3 displays the offline map of the University of Ilorin, highlighting key landmarks across the campus to estimate the distance between two locations.

2.4.1. Route definition and optimal pathfinding

Navigation from a user-defined start to the destination was performed using Dijkstra's Algorithm on the campus graph to compute the shortest-weight path, as presented in (10). The system then translated this optimal route into step-by-step directions, similar to standard GPS prompts (e.g., *"After 20 m, turn north*").

$$dist[v] = \min(dist[v], dist[u] + w(u, v)) \tag{10}$$

Where node (v) is the current shortest known campus route from the source to the destination, node (u) is the shortest known campus route to a location (e.g., hostels, lecture halls, library), and weight (w) is the distance between two points, computed using the Haversine formula.

Navigation begins with the user starting a destination via the SPH0645LM4H microphone. The speech is transcribed by Vosk, an offline, Raspberry Pi-friendly speech-to-text engine included in the system's directory. The system then maps the recognized location to campus coordinates and feeds them into the pathfinding algorithm to compute the optimal route. Afterwards, the navigation instructions are first generated as text (e.g., "Proceed straight for 50 m, then turn left") and converted to speech using the eSpeak TTS engine, delivered via earphones. To avoid overwhelming the user, updates are triggered incrementally based on GPS (NEO-6M) location changes, prompting the next instruction after each path segment is completed.

2.5. The CAN prototype development

The CAN prototype's hardware, shown in Figure 4, is built around the Raspberry Pi model B with 8GB of RAM, selected for its low-power AI processing capabilities. Key peripherals include a Pi camera v2, an HC-SR04 ultrasonic sensor, a speaker, and a GPS module, enabling object detection, obstacle detection, audio feedback, and offline navigation. Components were selected for affordability, efficiency, and portability, suitable for wearable technology. The system is estimated to consume about 14.5 W/hour and is powered by a 20 Ah, 63 Wh battery, ensuring at least 4 hours of continuous use.

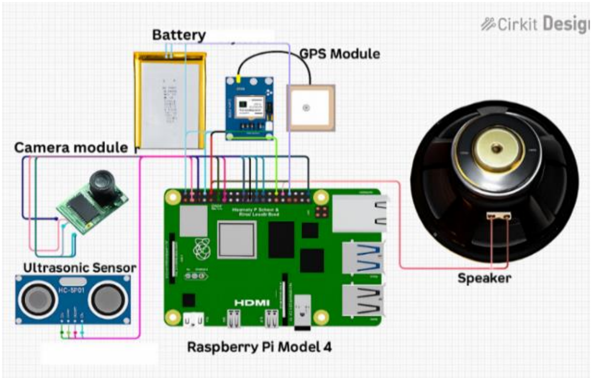

Figure 4. CAN device circuit implementation

Upon activating the CAN device, the user specifies the destination. The camera scans the surroundings, and nearby visuals are vocalised via the speaker before navigation begins. As the user moves, the system detects obstacles in real time and provides audio cues at GPS checkpoints to ensure route alignment, as outlined in the flowchart in Figure 5. Figure 6 shows the CAN device prototype, housed in lightweight, easy-to-shape foamboard for wearable use. Its design ensures comfort, with a fit similar to a face mask.

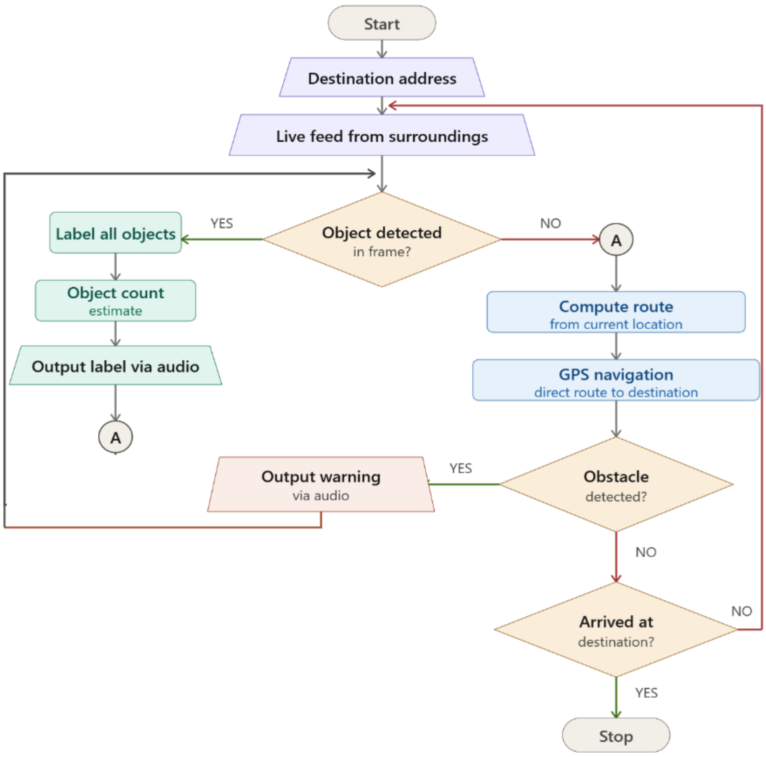

Figure 5. Flowchart of CAN device processes

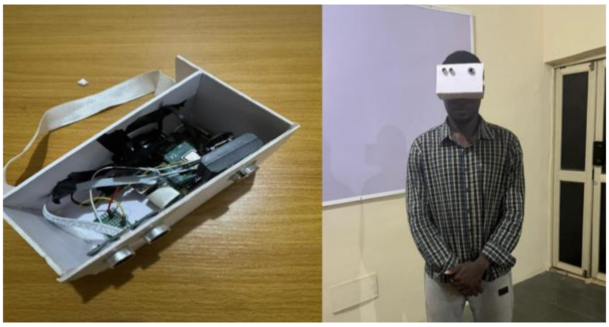

Figure 6. Packaging of the CAN device prototype

3. RESULTS AND DISCUSSION

The CAN device was evaluated on three key functions: object recognition, obstacle distance measurement, and navigation. Each module was tested separately, then combined and assessed as a complete wearable system. The training process supporting the object recognition module was configured with 100 epochs, a batch size of 32, and a learning rate of 0.001, ensuring stable convergence and reliable performance across the evaluation tasks.

3.1. Object recognition performance

The YOLO11-n model, trained on the University of Ilorin dataset, demonstrated strong performance in both training and testing. As shown in Figure 7, the consistent decline in loss values alongside rising mAP reflects effective learning and generalization. Its accuracy, efficiency, and rapid convergence highlight its suitability for edge deployment on the Raspberry Pi 4, ensuring precise detection and robust performance on unseen data.

During training, YOLO11-n exhibited consistent convergence across all loss functions. The box loss (Figure 7(a), $L_{\text{box}}(t)$) dropped from ~0.08 to ~0.01, showing improved bounding-box localization as expressed in (11). The objectness loss (Figure 7(b), $L_{\text{obj}}(t)$) decreased from ~0.025 to ~0.005, reflecting stronger confidence in object detection as shown in (12). The classification loss, as depicted in (13) (Figure 7(c), $L_{\text{cls}}(t)$) fell from ~0.02 to <0.002, indicating enhanced labeling precision. Each loss function decreases monotonically as the training progresses.

$$\frac{dL_{\text{box}}}{dt} < 0, \qquad L_{\text{box}}(t): 0.08 \to 0.01 \tag{11}$$

$$\frac{dL_{\text{obj}}}{dt} < 0, \qquad L_{\text{obj}}(t): 0.025 \to 0.005 \tag{12}$$

$$\frac{dL_{\text{cls}}}{dt} < 0, \qquad L_{\text{cls}}(t): 0.02 \to 0.002 \tag{13}$$

The total training objective combines these components, as expressed in (14):

$$L_{\text{total}}(t) = \alpha \cdot L_{\text{box}}(t) + \beta \cdot L_{\text{obj}}(t) + \gamma \cdot L_{\text{cls}}(t) \tag{14}$$

where α, β, γ are weighting coefficients balancing localization, detection confidence, and classification accuracy. As training progresses, the total loss converges as shown in (15).

$$\lim_{t \to T} L_{\text{total}}(t) \to L_{\text{min}} \tag{15}$$

YOLO11-n demonstrated strong training and generalization performance. Figure 7(d): precision reached ~0.95, indicating few false positives, as expressed in (16), while Figure 7(e): recall approached 1.0, confirming minimal missed detections, as expressed in (17), where TP, FP, and FN denote true positive, false positive, and false negative detections, respectively. On the validation data, Figure 7(f) shows that the box loss decreased from 0.025 to <0.01, and Figure 7(g) shows that the objectness loss decreased from 0.017 to ~0.005, indicating reliable localization and detection on unseen samples, as shown in (18) and (19), respectively:

$$P = \frac{TP}{TP+FP} \approx 0.95 \tag{16}$$

$$R = \frac{TP}{TP+FN} \rightarrow 1.0 \tag{17}$$

$$\frac{dL_{\text{val,box}}}{dt} < 0, \qquad L_{\text{val,box}}(t): 0.025 \rightarrow 0.01 \tag{18}$$

$$\frac{dL_{\text{val,obj}}}{dt} < 0, \qquad L_{\text{val,obj}}(t): 0.017 \rightarrow 0.005 \tag{19}$$

The total validation objective, obtained by combining (18) and (19), is given by (20). As training progresses, the total loss on the validation data converges, as shown in (21).

$$L_{\text{val,total}}(t) = \alpha \cdot L_{\text{val,box}}(t) + \beta \cdot L_{\text{val,obj}}(t) \tag{20}$$

$$\lim_{t \to T} L_{\text{val,total}}(t) \rightarrow L_{\text{min}} \tag{21}$$

YOLO11-n-nano's validation performance confirms robust classification and detection accuracy. The validation classification loss (Figure 7(h), $L_{\text{val,cls}}(t)$) shows a fluctuating but downward trend, stabilising near 0.002, indicating reliable classification on unseen data, as expressed in (22). The mAP at intersection over union (IoU) 0.5 (Figure 7(i), $\text{mAP}_{0.5}$) quickly exceeds 0.9, as shown in (23), reflecting high detection accuracy under lenient overlap requirements, while (Figure 7(j), stricter mAP@0.5:0.95) stabilises around 0.8 and above, as expressed in (24), demonstrating strong localization and classification even under challenging thresholds.

$$\frac{dL_{\text{val,cls}}}{dt} < 0, \qquad L_{\text{val,cls}}(t) \rightarrow 0.002 \tag{22}$$

$$\text{mAP}_{0.5} > 0.9 \tag{23}$$

$$\text{mAP}_{0.5:0.95} \approx 0.8 + \tag{24}$$

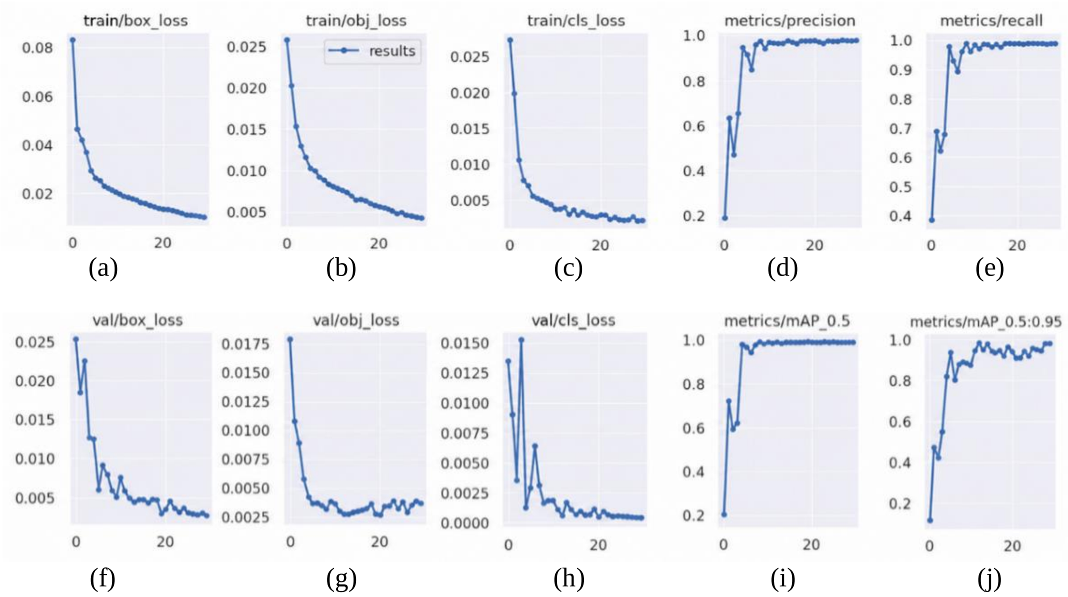

Figure 7. YOLO11-n training graph showing performance metrics across epochs: (a) training box loss, (b) training objectness loss, (c) training classification loss, (d) precision, (e) recall, (f) validation box loss, (g) validation objectness loss, (h) validation classification loss, (i) mAP@0.5, and (j) mAP@0.5:0.95.

In summary, YOLO11-n-nano learned and generalized well, with smoothly converging losses and no overfitting. Precision (~0.95), recall (~1.0), and consistently high mAP scores (>0.9 at IoU 0.5, >0.8 at stricter thresholds) confirm robust localization and classification, and reliable real-time detection in challenging conditions. Figure 8 shows labeled detections displayed on a monitor via Raspberry Pi, while Figure 9 illustrates successful object recognition at 55% confidence in low-light conditions.

Figure 8. Real-time object detection on Raspberry Pi with bounding boxes and class labels in bright conditions

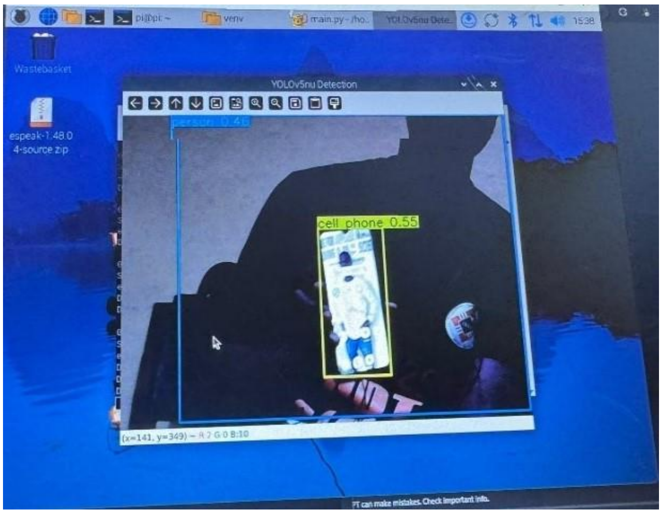

Figure 9. Real-time object detection on Raspberry Pi with bounding boxes and class labels in low-light conditions

3.2. Obstacle distance measurement

The HC-SR04 ultrasonic sensor was calibrated with minimal error (<2%) and closely matched tape-measured distances (Figure 10). Accurate readings are vital for obstacle avoidance. During tests, the feedback system functioned reliably: speech alerts announced objects detected at distances of 100 to 300 cm, while obstacles closer than 100 cm activated variable-frequency beeps in real time.

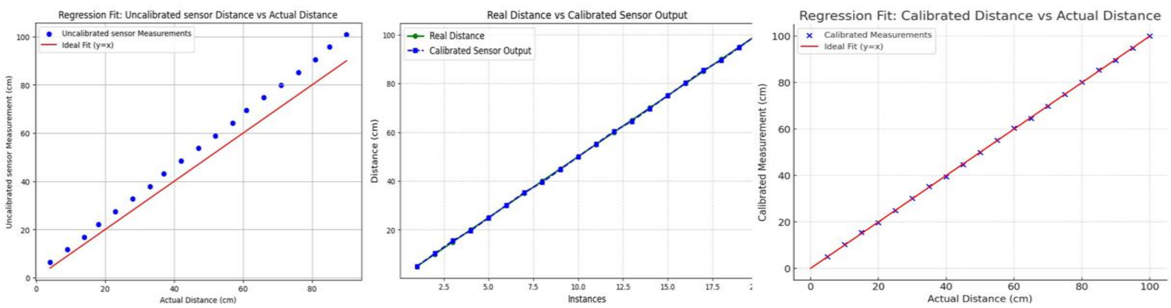

Figure 10. Distance measurements comparison between the measuring tape and the sensor output

3.3. Navigation assistance

Live tests showed that the NEO-6M GPS module achieved a ±1.5 m accuracy, which is sufficient for guiding users between campus landmarks. eSpeak-generated voice commands like “Proceed 30 m forward, then turn north,” offered clear, step-by-step navigation without overload. This confirms the effectiveness of offline map preprocessing and lightweight speech synthesis for reliable, internet-free navigation. Using custom OpenStreetMap data for the University of Ilorin, landmarks were mapped as graph nodes, enabling efficient routing via Dijkstra’s Algorithm. Figure 11 illustrates sample paths, with red lines marking pedestrian routes. As users moved, GPS points were logged in the Raspberry Pi’s memory and later visualised as shown in Figure 11. The final prototype integrated all modules into a compact wearable unit powered by a 20 Ah USB power bank (Figure 6). In field tests, it reliably detected obstacles, delivered real-time alerts, and guided users with clear audio prompts, boosting safety and independent navigation.

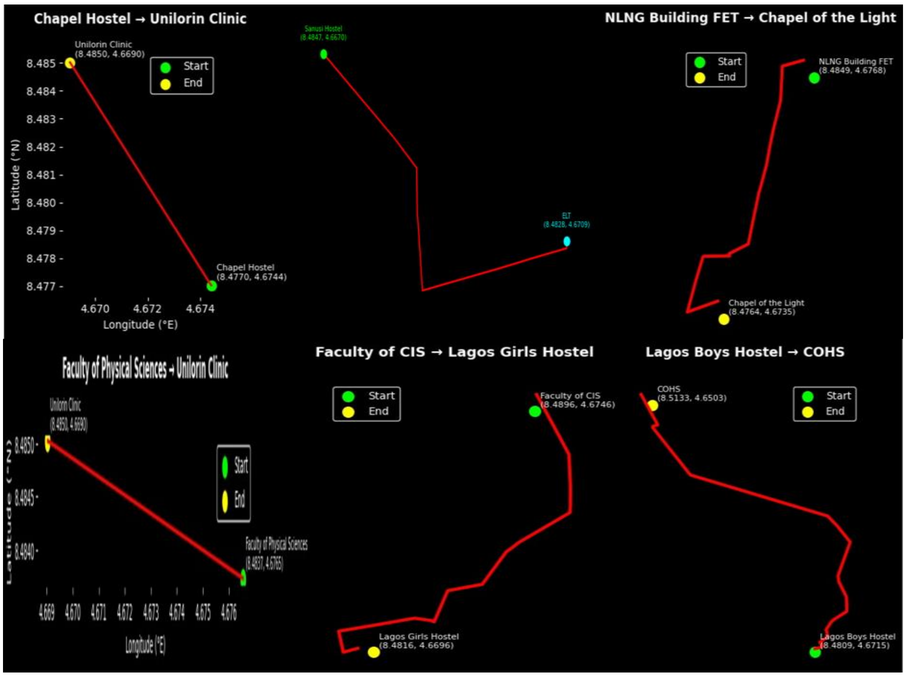

Figure 11. Pedestrian paths at different instances of testing

4. CONCLUSION

The CAN system effectively achieved its objectives of real-time object detection, adaptive obstacle avoidance, and offline navigation for visually impaired users. Its compact, wearable design demonstrates the feasibility of an affordable, internet-free assistive solution, laying a solid foundation for future development. However, challenges remain. Object detection accuracy dropped in low light due to camera limitations, and GPS performance was unreliable indoors. The lack of field testing with visually impaired users limits insights into real-world usability. Future improvements should include low-light cameras, more compact hardware, and enhanced navigation using GPS combined with inertial measurement units (IMUs) or indoor positioning methods such as bluetooth low energy (BLE), WiFi, or light detection and ranging (LiDAR). Incorporating natural-language audio feedback and conducting user trials will be essential to refining the system and boosting its effectiveness. Additionally, the device needs improvements in size, weight balance, and power efficiency for prolonged use.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mahmud Hafeez Owolabi	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓
Idajili John Ojochegbe	✓	✓	✓	✓		✓	✓	✓	✓		✓			
Ayinla Shehu Lukman				✓	✓					✓				✓
Jimoh-Mahmud Aishat Oladayo										✓				✓
Yusuf Abdulrahman Olalekan					✓					✓				✓
Olaogun Jerry Oluwajomiloju		✓	✓			✓	✓	✓						

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state there is no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, Mahmud, H. O., on reasonable request.

REFERENCES

[1] D. Abdulhussein and M. A. Hussein, "WHO vision 2020: have we done it?," *Ophthalmic Epidemiology*, vol. 30, no. 4, pp. 331–339, Jul. 2023, doi: 10.1080/09286586.2022.2127784.

[2] R. C. Joshi, S. Yadav, M. K. Dutta, and C. M. Travieso-Gonzalez, "Efficient multi-object detection and smart navigation using artificial intelligence for visually impaired people," *Entropy*, vol. 22, no. 9, p. 941, Aug. 2020, doi: 10.3390/e22090941.

[3] World Health Organization, "World report on vision," 2019. Accessed: Dec. 08, 2025. [Online]. Available: https://www.who.int/docs/default-source/documents/publications/world-vision-report-accessible.pdf

[4] T. Sarkar, A. Patel, and S. P. Arjunan, "Design and development of a smart eye wearable for the visually impaired," in *Information, Communication and Computing Technology. ICICCT 2020. Communications in Computer and Information Science*, C. Badica, P. Liatsis, L. Kharb, and D. Chahal, Eds., Singapore: Springer, 2020, pp. 208–221. doi: 10.1007/978-981-15-9671-1_18.

[5] P. Patel, S. Pampaniya, A. Ghosh, R. Raj, D. Karuppaih, and S. Kandasamy, "Enhancing accessibility through machine learning: A review on visual and hearing impairment technologies," *IEEE Access*, vol. 13, pp. 33286–33307, 2025, doi: 10.1109/ACCESS.2025.3539081.

[6] S. Jeeva, A. Nalini, T. Kaviya, P. Monisha, R. Kaviya, and S. Mathumathi, "Object detection system using YOLO model for blind people," in *2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, IEEE, Sep. 2025, pp. 220–226. doi: 10.1109/ICIMIA67127.2025.11200546.

[7] X. Li, J. Wang, F. Xu, and J. Song, "Improvement of YOLOv3 algorithm in workpiece detection," in *2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, IEEE, Jul. 2019, pp. 1063–1068. doi: 10.1109/CYBER46603.2019.9066490.

[8] B. Kuriakose, R. Shrestha, and F. E. Sandnes, "Tools and technologies for blind and visually impaired navigation support: a review," *IETE Technical Review*, vol. 39, no. 1, pp. 3–18, Jan. 2022, doi: 10.1080/02564602.2020.1819893.

[9] D. F. Jasim and W. F. Shareef, "Non-invasive tongue-based HCI system using deep learning for microgesture detection," *Revue d'Intelligence Artificielle*, vol. 37, no. 04, pp. 985–995, Aug. 2023, doi: 10.18280/ria.370420.

[10] D. F. Jasim and W. F. Shareef, "Ululate: a non-intrusive, wearable tongue gesture detection system for human-computer interaction," *Mathematical Modelling of Engineering Problems*, vol. 11, no. 1, pp. 263–272, Jan. 2024, doi: 10.18280/mmep.110129.

[11] M. Mukhiddinov and J. Cho, "Smart glass system using deep learning for the blind and visually impaired," *Electronics*, vol. 10, no. 22, p. 2756, Nov. 2021, doi: 10.3390/electronics10222756.

[12] C. de Paz, J. A. Huertas, J. Ibáñez-Gijón, J. A. Martín-Gonzalo, A. B. Varas, and D. Travieso, "Enhancing navigation and obstacle avoidance with a vibrotactile device as secondary electronic travel aid," *Disability and Rehabilitation: Assistive Technology*, vol. 20, no. 4, pp. 1140–1150, May 2025, doi: 10.1080/17483107.2024.2417264.

[13] Y. Wang, Z. Xing, L. Ma, A. Qu, and J. Xue, "Object detection algorithm for Lingwu long jujubes based on the improved SSD," *Agriculture*, vol. 12, no. 9, p. 1456, Sep. 2022, doi: 10.3390/agriculture12091456.

[14] M. Alruwaili *et al.*, "Deep learning-based YOLO models for the detection of people with disabilities," *IEEE Access*, vol. 12, pp. 2543–2566, 2024, doi: 10.1109/ACCESS.2023.3347169.
[15] T. B. Athallah, Z. Darojah, and A. Risnumawan, "Enhancing object detection and navigation for visually impaired individuals: a YOLOv3-based approach with GPS integration," in *2023 International Conference on Artificial Intelligence Robotics, Signal and Image Processing (AIRoSIP)*, IEEE, Aug. 2023, pp. 263–268. doi: 10.1109/AIRoSIP58759.2023.10874026.
[16] L. Zhao and S. Li, "Object detection algorithm based on improved YOLOv3," *Electronics*, vol. 9, no. 3, p. 537, Mar. 2020, doi: 10.3390/electronics9030537.
[17] S. N. Katkade, R. R. Manza, and C. Pattebahadur, "YOLOv5-based object detection system for visually impaired individuals using Raspberry Pi," *Artificial Intelligence and Applications*, vol. 4, no. 2, pp. 256–264, Jul. 2025, doi: 10.47852/bonviewAIA52024434.
[18] R. S. Lakshmi, M. Sneha, K. Swetha, and A. Thilagavathy, "AI-powered smart glasses for blind, deaf, and dumb," in *2022 5th International Conference on Advances in Science and Technology (ICAST)*, IEEE, Dec. 2022, pp. 280–285, doi: 10.1109/ICAST55766.2022.10039557.
[19] G. P. Kela, M. G. Daga, and R. Khandelwal, "Vision to voice: An advanced blind assistance system integrating YOLOv3 and OCR technologies for enhanced mobility," in *2025 3rd International Conference on Disruptive Technologies (ICDT)*, IEEE, Mar. 2025, pp. 1473–1476, doi: 10.1109/ICDT63985.2025.10986482.
[20] T. Sugashini and G. Balakrishnan, "YOLO glass: video-based smart object detection using squeeze and attention YOLO network," *Signal, Image and Video Processing*, vol. 18, no. 3, pp. 2105–2115, Apr. 2024, doi: 10.1007/s11760-023-02855-x.
[21] V. V. Meshram, K. Patil, V. A. Meshram, and F. C. Shu, "An astute assistive device for mobility and object recognition for visually impaired people," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 5, pp. 449–460, Oct. 2019, doi: 10.1109/THMS.2019.2931745.
[22] K. Makanyadevi, M. Avinash, V. S. Chandhrakiran, S. Kavitha, A. S. B. Krishna, and S. Dhiyanesh, "AI-powered smart glasses and shoes for the visually impaired," in *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, IEEE, Jan. 2025, pp. 1595–1603. doi: 10.1109/ICMCSI64620.2025.10883641.
[23] K. Kim, J. Kim, H.-G. Lee, J. Choi, J. Fan, and J. Joung, "UAV chasing based on YOLOv3 and object tracker for counter UAV systems," *IEEE Access*, vol. 11, pp. 34659–34673, 2023, doi: 10.1109/ACCESS.2023.3264603.
[24] A. S. Rao *et al.*, "Wearable assistive device for enhanced navigation in visually impaired individuals using YOLO-based object detection," in *2025 Fifth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, IEEE, Jan. 2025, pp. 1–7, doi: 10.1109/ICAECT63952.2025.10958989.
[25] D. Chaudhary, A. Mathur, A. Chauhan, and A. Gupta, "Assistive object recognition and obstacle detection system for the visually impaired using YOLO," in *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, Jan. 2023, pp. 353–358, doi: 10.1109/Confluence56041.2023.10048808.
[26] E. H. Alkhammash, "Multi-classification using YOLOv11 and hybrid YOLO11n-MobileNet models: a fire classes case study," *Fire*, vol. 8, no. 1, p. 17, Jan. 2025, doi: 10.3390/fire8010017.
[27] H. O. Mahmud, S. L. Ayinla, A. O. Jimoh-Mahmud, and A. O. Ekogiawe, "Development of a pattern recognition system for interpreting sign language using convolutional neural network," *Nigerian Journal of Technological Development*, vol. 22, no. 1, pp. 193–200, 2025, doi: 10.4314/njtd.v22i1.3365.

BIOGRAPHIES OF AUTHORS

Mahmud Hafeez Owolabi earned his B.Eng. (2013) and M.Eng. (2021) degrees in Electrical and Electronic Engineering from the University of Ilorin, Nigeria, and is currently pursuing a PhD in the same field at the University of Ibadan. He joined the University of Ilorin in 2014 as a technologist and transitioned to a lecturer role in 2023, where he now serves in the Department of Computer Engineering. His research interests encompass artificial intelligence of things (AIoTs), instrumentation and control, machine learning and deep learning algorithms, embedded systems and microprocessor architecture, wireless sensor networks, wearable devices, metaheuristic optimization algorithms, and renewable energy infrastructure. He can be contacted at email: mahmud.ho@unilorin.edu.ng.

Idajili John Ojochegbe recently graduated with a Bachelor's degree in Computer Engineering from the University of Ilorin, Nigeria, and his research interests include computer vision (CV), machine learning, robotics, and image/video processing with applications in automation and AIoT. He has gained experience as a CV Research Intern at Queryfier LLC, Colorado, where he worked on vision-language models for human orientation and object detection, and currently serves as a Research Assistant with TU Delft's Intelligent Vehicles Group (Department of Cognitive Robotics), Netherlands, focusing on multimodal perception for autonomous systems through audio-vision model integration for occluded object detection. He can be contacted at email: johnidajili9@gmail.com.

Ayinla Shehu Lukman received the B.Eng. in Electrical, Electronics and Computer Engineering and the M.Eng. in Electronic and Computer Engineering from Nnamdi Azikiwe University, Awka, Nigeria, in 2006 and 2017, respectively. He earned the Ph.D. in Electrical and Electronic Engineering from Universiti Teknologi PETRONAS, Perak, Malaysia, in 2026. From 2008 to 2013, he worked as a Nokia Maintenance Engineer at Best Service Raya Distribution Nigeria Limited. Since 2014, he has been affiliated with the University of Ilorin, Ilorin, Nigeria, where he is currently a Lecturer in the Department of Computer Engineering. His research interests include the Industrial Internet of Things (IIoT), indoor localization and positioning systems, machine learning, deep learning, explainable artificial intelligence (XAI), wireless sensor networks, wearable sensors, and engineering applications of optimization algorithms. He is a registered member of the Nigerian Society of Engineers (NSE), the Council for the Regulation of Engineering in Nigeria (COREN), and the IEEE Computer Society. He can be contacted at email: ayinla.sl@unilorin.edu.ng.

Jimoh-Mahmud Aishat Oladayo received her B.Tech. in Computer Engineering from Ladoke Akintola University of Technology, Nigeria, and her M.Tech. in Network Engineering from Manipal University, India. She is currently a Lecturer at University of Ilorin, Nigeria, and she is a registered member of the Council for the Regulation of Engineering in Nigeria (COREN). Her research focuses on intrusion detection, millimeter -wave communication, AI, and data analysis, with several publications contributing to secure and intelligent communication systems. Her academic work demonstrates a strong commitment to bridging theory with practical applications in emerging technologies. She can be contacted at email: jimohmahmud.oa@unilorin.edu.ng.

Yusuf Abdulrahman Olalekan received his B.Eng. in Electrical and Electronics Engineering from Federal University of Technology, Minna, and his M.Eng. from the Department of Electrical and Electronics Engineering, University of Ilorin, Nigeria. He has published over 20 peer-reviewed research articles, which have collectively received more than 60 citations on Google Scholar. His research interests include wearable biosensors, machine learning, and embedded system design. He can be contacted at email: yusuf.oa@unilorin.edu.ng.

Olaogun Jerry Oluwajomiloju is a first-class Computer Engineering graduate from the University of Ilorin, Nigeria, with interests in computer vision, embedded AI, and assistive perception systems. His work includes developing a smart glasses platform for real-time obstacle detection, object recognition, and navigation support for the visually impaired using camera vision models and multimodal sensors. He has practical technical experience from roles at Tunde Idiagbon International Airport, Ilorin (ICT/Communication Intern), where he supported server and network communication systems, and Softrays IT Solutions (Cybersecurity Intern), where he worked on threat monitoring, vulnerability assessment, and system hardening. He also brings product and interface design experience from XIRCLS (India) as a design intern, 1go Technology (design lead), and nniDialogue Institute (volunteer designer), contributing to UI layouts, brand systems, and digital product assets. He can be contacted at email: jerryolaogun@gmail.com.