

Development of An Object Detection System for the Visually Impaired

OLUWASEUN ADENIYI OJERINDE¹, RAMATU ABUBAKAR², ABUBAKRY KAREEM-OJO³,
ENOCH MIDA⁴

^{1,2,3,4} *Department of Computer Science, Federal University of Technology, Minna, Nigeria*

Abstract- *This study presents the development of an Android-based object detection application designed to assist visually impaired persons in recognizing and interacting with objects in their immediate environment. The system operates on an image processing and deep learning framework implemented using Google's TensorFlow API and Convolutional Neural Networks (CNN). The application allows users to capture real-time images through a mobile device camera, where detected objects are processed and conveyed to the user through synthesized voice output. Android was selected as the development platform due to its open architecture and built-in accessibility features, including speech recognition and text-to-speech services, which support intuitive and hands-free user interaction. The methodology adopted in this work included the review and definition of relevant concepts in pattern recognition and computer vision, evaluation of existing object detection applications to identify gaps, design of an improved assistive detection architecture, and implementation of the proposed system. Additional assistive modules such as Optical Character Recognition (OCR), voice assistance dialogue support, and image labeling using Firebase ML Kit were integrated to expand user awareness and usability. The model was evaluated using the TensorFlow Object Detection API, where a confidence threshold was applied to interpret detection reliability. Detection outputs with a confidence level of 60% and above were classified as accurate, while results below 59% were regarded as unreliable, ensuring dependable real-time feedback. Overall, the resulting system, referred to as Vocal Vision, enhances environmental awareness and independence for visually impaired users by providing real-time object detection, voice-guided navigation, and contextual understanding of their surroundings through a portable and user-friendly mobile interface.*

Index Terms- *Object Detection, Visually Impaired Assistance, TensorFlow, Convolutional Neural Network (CNN), Voice Feedback, Optical Character Recognition (OCR)*

I. INTRODUCTION

Visual impairment continues to be a major global health concern, affecting daily independence, mobility, communication, and social inclusion. According to the World Health Organization more than 2.2 billion people worldwide live with vision impairment or blindness, with many individuals facing substantial barriers in navigating physical environments [1]. The loss of vision often results in increased reliance on others for everyday tasks, and this dependence can significantly reduce quality of life[2]. As a result, developing effective assistive technologies that can compensate for visual limitations is a critical research and humanitarian priority[3].

Recent advances in mobile computing, artificial intelligence (AI), and computer vision have led to new opportunities to support visually impaired individuals. Modern smartphones are equipped with high-resolution cameras, powerful processors, and built-in accessibility features such as text-to-speech and voice recognition, making them suitable platforms for assistive applications [4]. Researchers have highlighted the significance of portable, affordable, and context-aware assistive tools, noting that smartphone-based solutions are more accessible than traditional, expensive standalone devices [5].

Deep learning, particularly Convolutional Neural Networks (CNNs), has transformed the field of object detection by enabling automated recognition of complex visual patterns [6]. Frameworks such as TensorFlow, developed by Google, have enabled efficient training and deployment of these models across multiple platforms [7]. For mobile deployment, TensorFlow Lite optimizes neural

network models to run efficiently on embedded systems and smartphones, enabling real-time object recognition in handheld devices[8]. Additionally, object detection algorithms such as YOLO (You Only Look Once) enable rapid image inference by processing the entire image in a single pass, making them suitable for real-time applications [9].

Assistive object recognition systems aim to detect objects in the environment and communicate their presence through audio feedback, thereby enhancing spatial awareness for visually impaired users[10]. However, many existing solutions face challenges, including limited detection accuracy, slow processing speed, poor usability, and lack of multi-feature support [4]. To address these issues, this study proposes Vocal Vision, an Android-based object detection and recognition system integrating TensorFlow, CNN-based object detection, YOLO-based real-time processing, Firebase ML Kit image labeling, and Optical Character Recognition (OCR) modules. The application captures images using the smartphone camera, detects objects, and provides voice-based verbal notification to the user, enabling hands-free environment awareness.

This study proposes “Vocal Vision,” an Android-based object detection application that employs the TensorFlow API alongside other modules (e.g., Optical Character Recognition, voice assistance, Firebase ML Kit) to detect, label, and verbally notify visually impaired users of objects in their environment. The system incorporates a confidence threshold ($\geq 60\%$ accurate detection) to ensure reliable feedback. By leveraging deep learning (e.g., CNNs) and mobile-optimized frameworks, this work advances the state of assistive technologies for visually impaired individuals by providing a portable, affordable, and accessible solution.

II. REVIEW OF RELATED LITERATURE

This section presents the review of related literature.

A. Assistive Technologies for the Visually Impaired
Assistive technology for visually impaired persons is designed to enhance independence, navigation, reading, and interaction with the environment [10]. Traditional tools such as the white cane and guide

dogs provide basic mobility assistance but are limited in detecting distant or elevated obstacles [11]. As a result, Electronic Travel Aids (ETAs) such as ultrasonic canes and tactile feedback devices were developed to offer additional spatial awareness, though many remain expensive or require specialized training [12].

Recent advances in artificial intelligence and mobile computing have led to the development of smartphone-based assistive systems. Applications such as Google Lookout, Seeing AI, and Envision AI use computer vision and machine learning models to recognize objects, read printed text, and describe scenes aloud to users [13]. These solutions are increasingly preferred because smartphones are affordable, portable, and widely available, reducing the need for specialized hardware [14]. Integrated features such as text-to-speech (TTS) and speech recognition further allow hands-free operation, which is essential for visually impaired users.

B. On-Device Object Detection Frameworks and Algorithms

Deep learning models especially Convolutional Neural Networks (CNNs) remain the backbone of modern object detection systems [15]. Frameworks such as TensorFlow and specialized mobile runtimes like TensorFlow Lite enable deployment of trained models on smartphones with practical latency and memory footprints [16]. The YOLO family (You Only Look Once), particularly recent lightweight variants (Mobile-YOLO, YOLOv8-Mobile), has been adapted for on-device, real-time inference because of its single-pass detection approach and favorable speed–accuracy tradeoffs [17]. Several studies report successful Android deployments using optimized YOLO and MobileNet-SSD backbones, demonstrating near real-time performance on commodity phones and acceptable accuracy for assistive tasks (Mobile-YOLO studies; GitHub YOLOv8-Mobile implementations [18]).

C. Tensor Flow Lite Versus Server-Side Interface

TensorFlow Lite (TFLite) enables machine learning models to run directly on mobile and embedded devices, making it suitable for offline and real-time applications [19]. By converting full TensorFlow

models into optimized, quantized versions, TFLite significantly reduces model size and computation requirements while maintaining acceptable accuracy [20]. This allows applications to run efficiently on smartphones, microcontrollers, and edge devices without the need for constant network connectivity [21]. For assistive technology used by visually impaired users, this independence from internet access is critical, as navigation tasks often need immediate, reliable feedback.

In contrast, server-side (cloud-based) inference involves sending input data such as images or audio to remote servers where more powerful GPU or TPU clusters perform the processing [22]. Cloud inference generally achieves higher accuracy and faster processing for large or complex models, such as high-resolution object detectors or large vision-language models [23]. However, this approach depends heavily on network quality. Latency, connectivity interruptions, and privacy risks particularly the transmission of live camera feeds can limit cloud-based systems in real-world mobility environments. For visually impaired users, even a delay of one second can result in unsafe navigation feedback.

D. Multi-Modal Assistive Features: OCR, Image Labeling and Voice

Practical Modern assistive technologies for visually impaired users increasingly rely on multi-modal interfaces that combine visual recognition, text reading, and auditory feedback [20]. Optical Character Recognition (OCR) enables mobile devices to extract text from printed materials such as books, signs, medicine labels, and currency [24]. Recent advancements in lightweight OCR engines, such as Firebase ML Kit OCR and Tesseract 5, allow real-time text extraction on smartphones without requiring cloud processing [25]. This feature is essential for visually impaired users, as it supports independent reading and enhances access to written information in public and private environments.

In addition to OCR, image labeling and scene description algorithms enhance user awareness of surroundings. These systems use deep convolutional neural networks to identify everyday objects,

landmarks, and activities in captured images, assigning labels that describe the visual environment. Lightweight object detection models, such as MobileNet-SSD and YOLOv8-Lite, have been optimized for mobile deployment, allowing real-time labeling on Android devices [26]. The integration of image labeling with environmental context enables visually impaired users to receive more meaningful guidance beyond simple obstacle detection.

E. Voice Interaction and Dialogue Systems

Voice interaction and dialogue systems enable users to communicate with digital devices through natural spoken language rather than visual or touch-based interfaces [27]. For visually impaired individuals, speech-based interaction is essential because it provides an intuitive and hands-free method of controlling applications and receiving information. Modern speech recognition technologies use deep learning models such as recurrent neural networks (RNNs) and transformer-based architectures to convert spoken words into text with high accuracy, even in noisy environments [28]. These systems allow users to perform tasks such as opening applications, selecting functions, or initiating object detection without relying on a graphical user interface.

Dialogue systems extend speech recognition by enabling conversational interaction between the user and the device. Frameworks such as Google Dialogflow, Amazon Lex, and Microsoft LUIS support natural language understanding (NLU) and intent recognition, allowing the application to interpret user commands and respond appropriately. Recent advances in context-aware dialogue management allow systems to retain conversation history, improving coherence and reducing user frustration during repeated interactions [29]. For visually impaired users, these capabilities make mobile applications more accessible, as they no longer need to remember rigid voice commands or navigate complex menu structures. Text-to-Speech (TTS) systems complement speech recognition and dialogue systems by generating auditory feedback that conveys results, instructions, or real-time scene descriptions. Modern TTS models, such as Google's Neural TTS and Meta's Voicebox, produce natural-

sounding speech with improved tone, rhythm, and emotion [30]. When integrated with OCR and object detection modules, voice feedback provides immediate and context-specific guidance, enhancing situational awareness and independence. Thus, voice interaction and dialogue systems play a central role in multi-modal assistive technologies by bridging machine perception and user communication.

F. Recent Applications and Case Studies

Rahman et al. present a smartphone system for seamless scene and object recognition in complex spaces [31]. Khadidos and Yafoz [32] proposed hybrid models targeted at improved accuracy for assistive recognition. Mužinić [27] provided a broad survey of assistive systems in sensors-oriented applications. These works show that combining lightweight detection models with domain-specific optimization and human-centred design yields practical assistive tools.

Despite strong progress, key challenges persist:

- 1) Robustness in varied real-world conditions: changes in lighting, occlusion, and viewpoint reduce accuracy[33].
- 2) Limited object classes: many systems detect only a small vocabulary of objects that limits general usefulness[31].
- 3) Latency vs. accuracy tradeoffs: mobile models must balance resource constraints with acceptable detection reliability (TensorFlow Lite comparisons; Mobile-YOLO studies).
- 4) User-centred evaluation: few studies report large-scale trials with visually impaired users in diverse real-world tasks, and human factors (notification timing, verbosity) remain under-explored [33].

III. METHODOLOGY

The methodology adopted for this study outlines the systematic procedures used in designing, developing, and evaluating an Android-based object detection application tailored for visually impaired users. The approach integrates machine learning, mobile software engineering, and assistive technology design principles.

A. System Architecture and Design

In this stage, the conceptual framework and data flow were defined. The design centered on a mobile edge-AI architecture where object detection occurs locally on the smartphone. The system architecture consists of five main components as depicted in Fig.1:

- 1) Camera module for real-time image acquisition
- 2) Pre-trained deep learning detection model
- 3) TensorFlow Lite inference engine
- 4) Audio feedback subsystem (Text-to-Speech) and
- 5) User interaction layer optimized for accessibility.

The architecture emphasizes efficiency, modularity, and compatibility with Android's open ecosystem, ensuring seamless integration of speech services, vibration cues, and minimal-touch interfaces.

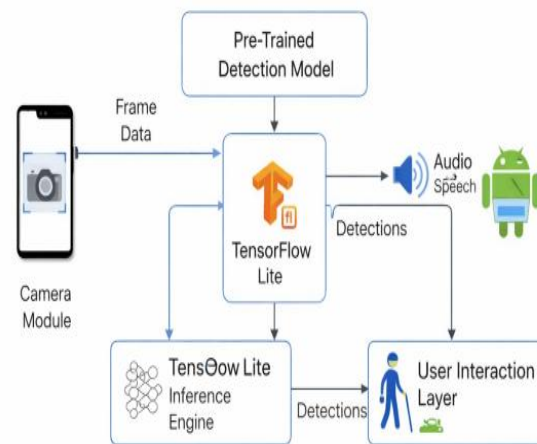


Fig. 1: System Architecture

The system design adopts a mobile edge-AI architecture in which object detection is performed entirely on the Android smartphone to ensure real-time performance and offline operation. The camera module captures continuous image frames, which are passed directly to the TensorFlow Lite inference engine hosting a pre-trained deep learning detection model. The inference engine processes each frame to identify objects and generate detection results with confidence scores. These results are then forwarded to the user interaction layer, where accessibility-oriented logic determines appropriate responses, and to the audio feedback subsystem, which uses text-to-speech (TTS) to verbally announce detected objects to the visually impaired user. Optional detection logs may be stored locally for performance analysis. This modular design ensures efficiency, low latency, and

seamless integration with Android's built-in accessibility services, including voice output, vibration cues, and minimal-touch interaction.

The system architecture is structured into layered components to ensure efficient, real-time object detection and accessible interaction for visually impaired users. At the Input Layer, the camera acquisition module utilizes the smartphone's rear camera to capture continuous video frames in real time using the CameraX or Camera2 API. These frames are streamed at a stable rate and undergo basic preprocessing operations such as resizing, rotation correction, and RGB normalization to match the input requirements of the detection model. The processed frames are then forwarded directly to the processing layer, ensuring smooth data flow and low latency suitable for real-time assistive applications. The Processing and Application Layers operate collaboratively to perform intelligent decision-making and user interaction. The processing layer hosts the optimized TensorFlow Lite inference engine, which executes the pre-trained object detection model on-device, producing bounding boxes, object class labels, and confidence scores for each detected object. The application logic layer then filters detections based on a defined confidence threshold, translates recognized object labels into natural language descriptions, and triggers appropriate feedback mechanisms. Finally, the Output Layer delivers results through accessibility-focused interfaces, primarily using text-to-speech (TTS) to announce detected objects, complemented by optional haptic feedback and minimal user interface elements. This layered design ensures offline functionality, protects user privacy, and provides a seamless, intuitive experience aligned with Android's accessibility ecosystem.

B. Dataset Development and Preparation

A hybrid dataset was constructed using publicly available datasets such as COCO and Open Images, complemented with custom images captured in real-world environments relevant to visually impaired users. Images were annotated using the LabelImg and Roboflow tools to generate bounding boxes for target object classes. Pre-processing steps included resizing images, normalization, and data augmentation (rotation, scaling, illumination adjustment) to improve

model robustness. The annotated dataset was converted to TFRecord format as required by the TensorFlow Object Detection API.

C. Model Training and Optimization

Model training was carried out using the TensorFlow Object Detection API. Models such as SSD MobileNet V2 and YOLO-based lightweight variants were considered due to their balance of speed and accuracy on mobile hardware. Training was performed on a GPU-enabled workstation with hyperparameters tuned for optimal convergence, including batch size, learning rate, and number of training steps. After training, models were converted to TensorFlow Lite format with quantization (INT8/Float16) applied to reduce model size and improve mobile inference speed. The final model outputs included bounding box coordinates, class labels, and confidence scores.

D. Android Application Development

The application was developed using Android Studio, integrating CameraX for continuous frame capture, the TensorFlow Lite Interpreter for on-device inference, and Android Text-to-Speech (TTS) for auditory feedback. Detected objects with confidence levels $\geq 60\%$ were classified as accurate detections, following experimental evaluation. Objects below the 59% threshold were flagged as unreliable and suppressed to minimize false announcements. The interface was designed with large buttons, high contrast, and minimal navigation steps, ensuring usability for visually impaired individuals. Additional features such as vibration alerts and automatic voice prompts enhanced accessibility.

IV. MATHEMATICAL MODELS

The proposed Android-based object detection system is mathematically modeled as a real-time visual perception pipeline that maps continuous image inputs to semantic object descriptions and audio feedback.

Let the system be represented as a function \mathcal{S} that transforms camera input into accessible outputs for visually impaired users.

1) Input Image Representation

Let a captured video frame from the smartphone camera be represented as in (1):

$$I_t \in \mathbb{R}^{H \times W \times C} \quad (1)$$

where:

H and W denote the image height and width,
 $C = 3$ represents RGB color channels,
 t denotes the time index of the video frame.

After preprocessing (resizing, normalization, and rotation correction), the input image is transformed into (2):

$$\hat{I}_t = \mathcal{P}(I_t) \quad (2)$$

where $\mathcal{P}(\cdot)$ denotes the preprocessing function.

2) Object Detection Model

The object detection model is defined as a deep neural network function as shown in (3):

$$\mathcal{F}_\theta: \mathbb{R}^{H' \times W' \times C} \rightarrow \mathcal{D} \quad (3)$$

where:

θ represents the learned model parameters,
 $H' \times W'$ are the resized input dimensions,
 \mathcal{D} is the set of detected objects.

For each frame \hat{I}_t , the model produces a set of detections:

$$\mathcal{D}_t = \{(b_i, c_i, s_i) | i = 1, 2, \dots, N\} \quad (4)$$

where:

$b_i = (x_i, y_i, w_i, h_i)$ is the bounding box of object i ,
 $c_i \in \mathcal{C}$ is the predicted object class label,
 $s_i \in [0, 1]$ is the confidence score,
 N is the total number of detected objects.

3) Confidence Filtering and Selection

To eliminate weak or false detections, a confidence

threshold τ is applied:

$$\mathcal{D}_t^* = \{(b_i, c_i, s_i) \in \mathcal{D}_t | s_i \geq \tau\} \quad (5)$$

Only detections with confidence scores greater than or equal to τ are forwarded to the application logic layer.

4) Semantic Mapping and Audio Feedback

Each detected class label is mapped to a natural language description using a mapping function:

$$L: \mathcal{C} \rightarrow \mathcal{W}$$

\mathcal{W}

where \mathcal{W} represents spoken words or phrases.

The audio output is generated using the Text-to-Speech (TTS) function in (6):

$$A_t = \mathcal{J}(L(c_i)) \quad (6)$$

where:

$\mathcal{J}(\cdot)$ is the TTS engine,

A_t is the synthesized audio signal delivered to the user.

5) End-to-End System Model

The complete system can be expressed as a composition of functions in (7):

$$A_t = \mathcal{J}(L(\mathcal{F}_\theta(\mathcal{P}(I_t)))) \quad (7)$$

This formulation demonstrates how raw camera input is transformed into meaningful, real-time audio feedback for visually impaired users.

6) Performance Constraints

Given mobile hardware limitations, inference latency T_{inf} must satisfy:

$$T_{inf} \leq T_{max} \quad (8)$$

to ensure real-time operation, where T_{max} is the maximum tolerable delay for assistive feedback.

The mathematical model formalizes the system as a real-time edge-AI pipeline that integrates image acquisition, deep learning-based object detection, confidence filtering, semantic interpretation, and accessible audio output. This model ensures efficient on-device processing, privacy preservation, and reliable assistive functionality for visually impaired users.

V. LOSS FUNCTION FORMULATION

The object detection model is trained by minimizing a multi-component loss function that jointly optimizes object localization, classification accuracy, and confidence prediction. Let the total loss \mathcal{L}_{total} be defined as in (9):

$$\mathcal{L}_{total} = \lambda_{loc} \mathcal{L}_{loc} + \lambda_{cls} \mathcal{L}_{cls} + \lambda_{conf} \mathcal{L}_{conf}$$

(9)

where:

- \mathcal{L}_{loc} is the localization loss,
- \mathcal{L}_{cls} is the classification loss
- \mathcal{L}_{conf} is the confidence (objectness) loss
- $\lambda_{loc}, \lambda_{cls}, \lambda_{conf}$ are weighting coefficients.

VI. LOCALIZATION LOSS

Localization loss measures the error between predicted bounding boxes and ground-truth boxes. It is defined using the Smooth L1 (Huber) loss as in (10):

$$\mathcal{L}_{loc} = \sum_{i=1}^N \text{SmoothL1}(b_i - \hat{b}_i) \quad (10)$$

where:

- $b_i = (x_i, y_i, w_i, h_i)$ is the predicted bounding box
- \hat{b}_i is the corresponding ground-truth bounding box,
- N is the number of matched anchor boxes.

The Smooth L1 function is defined as (11):

$$\text{SmoothL1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (11)$$

VII. CLASSIFICATION LOSS

The classification loss penalizes incorrect object class predictions. A categorical cross-entropy loss is used:

$$\mathcal{L}_{cls} = - \sum_{i=1}^N \sum_{c=1}^C \hat{y}_{ic} \log(y_{ic}) \quad (12)$$

where:

- C is the total number of object classes,
- \hat{y}_{ic} is the ground-truth label (one-hot encoded),
- y_{ic} is the predicted class probability.

VIII. CONFIDENCE (OBJECTNESS) LOSS

The confidence loss evaluates whether an object exists within a predicted bounding box. Binary cross entropy loss is applied:

$$\mathcal{L}_{conf} = - \sum_{i=1}^N [\hat{o}_i \log(o_i) + (1 - \hat{o}_i) \log(1 - o_i)] \quad (13)$$

where:

- o_i is the predicted objectness score,
- $\hat{o}_i \in \{0,1\}$ indicates the presence of an object.

IX. OPTIMIZATION OBJECTIVE

The training process seeks to minimize the total loss:

$$\theta^* = \underset{\theta}{\text{argmin}} \mathcal{L}_{total} \quad (15)$$

where θ represents the learnable parameters of the detection model. Optimization is performed using gradient-based methods such as Adam or stochastic gradient descent (SGD).

The loss function formulation ensures accurate object localization, correct classification, and reliable confidence estimation. By jointly optimizing these objectives, the trained model achieves high detection accuracy while maintaining real-time performance suitable for on-device inference on Android smartphones.

X. EXPERIMENTS AND RESULTS

The system was implemented using Python programming language on the Anaconda Integrated Development Environment. Various built-in modules were employed during coding and additional modules and libraries were downloaded and used, these libraries include, tensorflow, Numpy and Keras. In this study, the TensorFlow Object Detection API was used to evaluate the model by running inference on test images and comparing the predicted outputs with the ground-truth annotations. For each detected object, the model outputs a class label (e.g., *person*), a bounding box, and a confidence score that represents the model's certainty about the detection. A confidence threshold was applied to interpret detection correctness. Detections with a confidence

score of 60% and above were considered accurate, meaning the model reliably identified the object class and location. Detections with a confidence score of 59% and below were considered inaccurate and were either ignored or treated as false positives. As illustrated in the sample output image, the model detected a *person* with a confidence of 79%, which exceeds the threshold and is therefore classified as a correct detection. This threshold-based evaluation helps balance precision and recall, ensuring that only reliable detections are communicated to visually impaired users through audio feedback.

1) Detection Accuracy Under Night-Time Conditions

Fig 1. and 2 illustrates the performance of the proposed object detection system in a day light and low-light (night-time) environment. Five television screens are detected in the scene, each enclosed by a bounding box and labeled “tv” with confidence scores of 63% and 60%, respectively. These confidence values are relatively close to the predefined threshold of 60%, indicating that the system is able to recognize objects even under poor illumination, though with reduced certainty compared to well-lit conditions. The reduced confidence scores highlight the challenges associated with night-time detection, such as insufficient lighting, glare from bright screens, and image noise, all of which affect feature extraction and model certainty. Despite these limitations, the system successfully identifies the objects within an average processing time of 527 ms and 460 ms, demonstrating its capability for near real-time inference on a mobile device.

This result shows that while detection accuracy slightly degrades in low-light scenarios, the model remains functional and usable, reinforcing its practicality for assistive applications that may be used at any time of day.

Table 1 presents the night-time detection performance of the system for persons and laptops. Out of five images tested for each object class, none were correctly detected with a confidence score above the 60% threshold. This result indicates that the proposed system performs poorly under low-light conditions, demonstrating limited effectiveness for

night-time object detection without additional illumination or enhancement techniques.

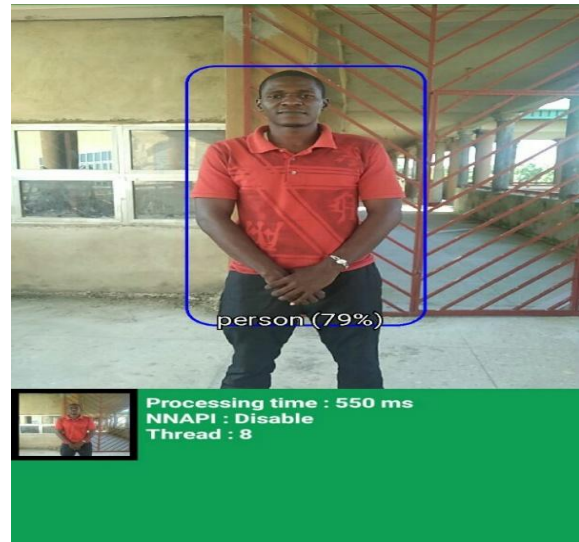


Fig 1. Performance of the proposed object detection system in a day-light

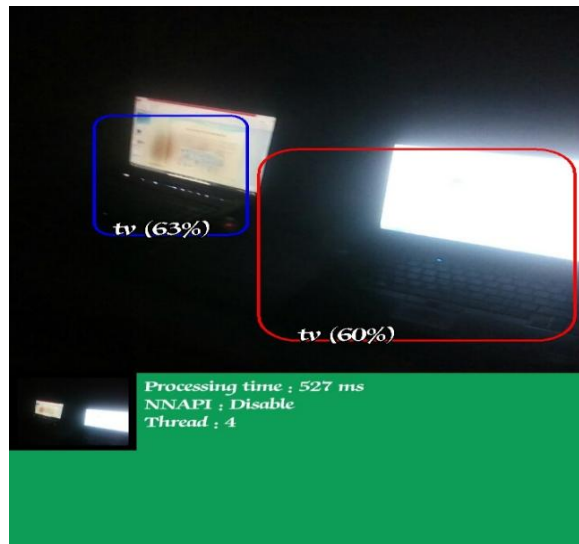


Fig 2. Performance of the proposed object detection system in the night

Table 1: Night-time detection performance of the system for persons and laptops

S/N	Object	Number of images	Correct Detect	True accuracy Above 60%
1	PERSONS	5	0	0

2	LAPTOPS	5	0	0
---	---------	---	---	---

Fig. 3 demonstrates the performance of the proposed object detection system in a typical classroom environment under normal lighting conditions. Multiple objects, including *persons*, *laptops*, and a *keyboard*, are correctly identified with confidence scores ranging from 60% to 71%, all of which meet or exceed the defined accuracy threshold. The presence of multiple bounding boxes indicates the system's ability to detect several objects simultaneously within a crowded scene. With an average processing time of 505 ms, the results show that the system performs effectively in indoor, well-lit environments, making it suitable for real-world assistive applications such as classroom navigation and object awareness for visually impaired users.

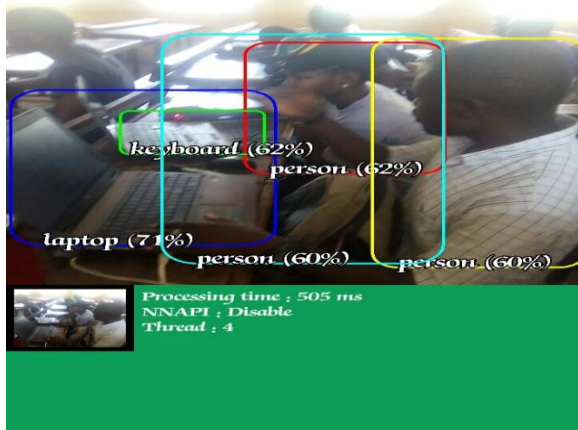


Fig. 3 Performance of the proposed object detection system in a typical classroom environment under normal lighting conditions.

Table 2 presents the detection accuracy of the system in a classroom environment. The results show that laptops and persons were correctly detected in all tested images, achieving average accuracy values of 69% and 75%, respectively, while keyboards were correctly detected in three out of five images with an accuracy of 62%. These findings indicate that the system performs reliably in well-lit classroom settings, confirming its effectiveness for indoor object detection and assistive support for visually impaired users.

Table 2. Classroom Accuracy Detection

S/ N	Objects	Number of images	Correct detected	True accuracy Above
1	LAPTOP	5	5	69%
2	PERSONS	5	5	75%
3	KEYBOAR D	5	3	62%

Fig. 4 illustrates the performance of the proposed object detection system in an outdoor footpath environment under natural daylight conditions. The system successfully detects multiple dynamic and static objects, including persons and cars, with confidence scores ranging from 62% to 76%, all of which exceed the predefined accuracy threshold. The presence of multiple bounding boxes across varying distances demonstrates the model's capability to handle crowded scenes and depth variation in outdoor settings. With processing times between 383 ms and 423 ms, the results indicate efficient real-time inference on a mobile device, confirming that the system is effective for outdoor navigation and situational awareness for visually impaired users.

Table 3 presents the object detection performance of the system in a footpath environment. All tested images containing persons and cars were correctly detected, achieving average accuracy values of 75% and 68%, respectively, which are above the defined confidence threshold. These results confirm that the system performs effectively in outdoor walking paths, supporting safe navigation and object awareness for visually impaired users.

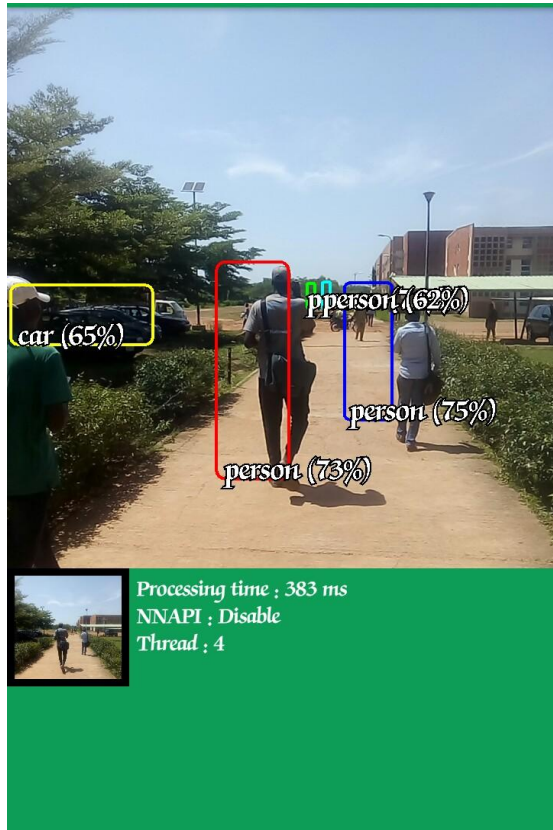


Fig. 4. Performance of the proposed object detection system in an outdoor footpath environment under natural daylight conditions.

Table 3. Footpath Detection Accuracy

S/N	Object	Number of images	Correct Detect	True accuracy Above 60%
1	PERSONS	5	5	75%
2	CARS	5	5	68%

XI. CONCLUSION

This research has shown that implementing convolutional neural network-based object detection on Android smartphones is both practical and effective for assistive applications designed for visually impaired users. By utilizing on-device inference with TensorFlow Lite, the system achieves acceptable detection accuracy and real-time performance without the need for expensive

REFERENCES

- [1] World Health Organization, Pneumonia. World Health Organization, 2023.
- [2] M.I. Ismail and M.A.A Mousa. Employing Computer Vision on a Smartphone to Help the Visually Impaired Cross the Road. Proceedings of the AAAI Symposium Series, 6(1), 2025. <https://doi.org/10.1609/aaais.v6i1.36057>
- [3] S.S. More, N. Patil, V.B. Lobo, N. Shet, D. Goswami and P. Rane. Empowering the Visually Impaired: YOLOv8-based Object Detection in Android Applications. Procedia Computer Science, 252, 2025. <https://doi.org/10.1016/j.procs.2025.01.005>
- [4] D.R. Pawar. Smart Vision: Real-Time Object Detection and Audio Assistance for the Visually Impaired Using TensorFlow and SSD MobileNet. International Journal for Research in Applied Science and Engineering Technology, 13(3), 2025. <https://doi.org/10.22214/ijraset.2025.6802>
- [5] A. Badave, R. Jagtap, R. Kaovasia, S. Rahatwad, and S. Kulkarni. Android Based Object Detection System for Visually Impaired. 2020 International Conference on Industry 4.0 Technology, I4Tech 2020. <https://doi.org/10.1109/I4Tech48345.2020.9102694>
- [6] J. Anitha, A. Subalaxmi and G. Vijayalakshmi. Real time object detection for visually challenged persons. International Journal of Innovative Technology and Exploring Engineering, 8(8), 2019.
- [7] A. Salunkhe, M. Raut, S. Santra, and S. Bhagwat, S. Android-based object recognition application for visually impaired. ITM Web of Conferences, 40, 2021. <https://doi.org/10.1051/itmconf/20214003001>
- [8] G. Khekare and K. Solanki. Real time object detection with speech recognition using tensorflow LITE. GIS Science Journal, 2022.
- [9] Y. Lei, C. Shan, W. Ge, X. Wang, X., and J. Hu. eMNV4-YOLO: A high-efficiency target detection framework for robotic driving vehicle

- instrument reading. *IEEE Access*, 13, 2025. <https://doi.org/10.1109/ACCESS.2025.3549298>
- [10] M. Mashilo and T. Iyamu, T. Selecting an assistive technology for visually impaired students in higher institution. *Issues in Information Systems*, 25(4), 2024. https://doi.org/10.48009/4_iis_2024_13.
- [11] S. Ikram, I. Sarwar Bajwa, S. Gyawali, A. Ikram and N. Alsubaie. Enhancing Object Detection in Assistive Technology for the Visually Impaired: A DETR-Based Approach. *IEEE Access*, 13, 2025. <https://doi.org/10.1109/ACCESS.2025.3558370>
- [12] Deshpande, V., Shelke, G., & Kadam, B. Empowering Vision: A survey on image captioning assistive technologies for the visually impaired. *Lecture Notes in Networks and Systems*, 1459 LNNS. 2026. https://doi.org/10.1007/978-981-96-7505-0_28
- [13] A.A. Adegun, J.V. Fonou-Dombeu, S. Viriri and J. Odindi . Ontology-based deep learning model for object detection and image classification in smart city concepts. *Smart Cities*, 7(4), 2024. <https://doi.org/10.3390/smartcities7040086>
- [14] W.F. Talafha,R.F. Bataineh. Breaking barriers: assistive technology for visually impaired EFL Educators. *International Journal of Learning, Teaching and Educational Research*, 24(4), 2025. <https://doi.org/10.26803/ijlter.24.4.38>
- [15] P. Ruiz-Barroso, F.M. Castro and N. Gui. (2025). Real-time unsupervised video object detection on the edge. *Future Generation Computer Systems*, 167, 2025. <https://doi.org/10.1016/j.future.2025.107737>
- [16] G. Fu, G. Gu, W. Liu AND H. Fu. . LISA-YOLO: A Symmetry-Guided Lightweight Small Object Detection Framework for Thyroid Ultrasound Images. *Symmetry*, 17(8), 2025. <https://doi.org/10.3390/sym17081249>
- [17] S.R. Kotha, S. Kanchanapalli, K. Gundaveni, H.B. Valiveti, B. Vandana, H. Muhamed, M., Rajput and A. Singla. Image captioning and speech synthesis in regional languages. *AIP Conference Proceedings*, 3263(1), 2025. <https://doi.org/10.1063/5.0261410>
- [18] J. Kunhoth, M. Alkaeed, A. Ehsan, A., and J. Qadir. VisualAid+: Assistive System for Visually Impaired with TinyML Enhanced Object Detection and Scene Narration. *International Symposium on Networks, Computers and Communications*, 2023.
- [19] Y. Yao, H. Liang, X. Li, J. Zhang and J. He. Sensing urban land-use patterns by integrating Google Tensorflow and scene-classification models. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(2W7), 2017. <https://doi.org/10.5194/isprs-archives-XLII-2-W7-981-2017>
- [20] H. Coy, K. Hsieh, W. Wu, N.B. Nagarajan, J.R. Young, M. L. Douek, M.S. Brown, F. Scalzo, F. and S.S.Raman. Deep learning and radiomics: the utility of Google TensorFlowTM Inception in classifying clear cell renal cell carcinoma and oncocytoma on multiphasic CT. *Abdominal Radiology*, 44(6), 2019. <https://doi.org/10.1007/s00261-019-01929-0>
- [21] M. Arya and G.H. Sastry. DEAL–‘Deep Ensemble Algorithm’ framework for credit card fraud detection in real-time data stream with google tensorflow. *Smart Science*, 8(2), 2022.
- [22] C.H. Hsieh, D.C. Lin, C.J. Wang, Z.T. Chen and J.J Liaw. Real-time car detection and driving safety alarm system with google tensorflow object detection API. *Proceedings - International Conference on Machine Learning and Cybernetics*, 2019.
- [23] S.R. Kotha,S. Kanchanapalli,K. Gundaveni, H.B. Valiveti,B. Vandana, M. Muhamed, M. Rajput and A. Singla). Image captioning and speech synthesis in regional languages. *AIP Conference Proceedings*, 3263(1),2025.<https://doi.org/10.1063/5.0261410>
- [24] F.A.Rahman, F. A., Pusparani, W.L., Yeoh and O., Fukuda, O. Smartphone-Based Seamless Scene and Object Recognition for Visually Impaired Persons. *Information (Switzerland)*, 16(9), 2025. <https://doi.org/10.3390/info16090808>
- [25] S. Zhao, Y. Zhang, Y. Wang, Z. Ren, P. Wei, T. Zhang, R. Peng, H. Zhou and F. Hu. Sample-to-answer nucleic acid detection using a fully

- integrated microdevice for nucleic acid extraction and smartphone-based droplet digital RPA/CRISPR. *Biosensors and Bioelectronics*, 289, 2025. <https://doi.org/10.1016/J.BIOS.2025.117886>
- [26] Y. Zhu, Y. Wang, Y. An, H. Yang, and Y. Pan. Real-Time Vehicle Detection And Urban Traffic Behavior Analysis Based On UAV Traffic Videos On Mobile Devices. arXiv preprint arXiv:2402.16246 2024.
- [27] M. Mužinić. A comparative study of deep learning-based text-to-speech approaches with an exploration of voice cloning techniques. (Doctoral dissertation, Sveučilište u Splitu, Sveučilište u Splitu, Prirodoslovno-matematički fakultet, Odjel za informatiku). 2025
- [28] T. Xie, Y. Rong, P. Zhang, W. Wang and L. Liu. Towards Controllable Speech Synthesis in the Era of Large Language Models: A Systematic Survey. 2024. <https://github.com/imxtx/>
- [29] E.A. Mohamed, A. Koura, and M. Kayed. Speech Emotion Recognition in Multimodal Environments with Transformer: Arabic and English Audio Datasets. *International Journal of Advanced Computer Science and Applications*, 15(3), 2024. <https://doi.org/10.14569/IJACSA.2024.0150359>
- [30] Y.O. Sharrab, H. Attar, M.A. Eljinini, Y. Al-Omary, and W.E. Al-Momani. Advancements in Speech Recognition: A Systematic Review of Deep Learning Transformer Models, Trends, Innovations, and Future Directions. In *IEEE Access*, 13, 2025
- [31] A. Sharma, A. Srivastava and A. Vashishth. An Assistive Reading System for Visually Impaired using OCR and TTS. *International Journal of Computer Applications*, 95(2), 2014. <https://doi.org/10.5120/16566-6231>
- [32] A.O. Khadidos and A. Yafoz. An intelligent object detection and classification framework for assisting visually challenged persons using deep learning and improved crowd search optimization. *Scientific Reports*, 15(1), 2025. <https://doi.org/10.1038/s41598-025-15793-0>
- [33] A.O. Khadidos and A. Yafoz. Intelligent guidance system for the visually impaired person using a lightweight YOLOv4-based object detection and sparse autoencoder framework. *Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engineers, Series A*, 2025. <https://doi.org/10.1080/02533839.2025.2538511>