Real-Time Object Detection App for Android with Voice Feedback

Dr.Shikha Gautam
Head Of Department
Dept.of Computer Engineering
Poornima Instituteof Engineeringand Technology , Jaipur
Shikha.gautam@poornima.org

Praveen Kumar
Student
Dept. of Computer Engineering
Poornima Instituteof Engineeringand Technology ,Jaipur
2021pietcspraveen130@poornima.org

Rishabh Choudhary
Student
Dept. of Computer Engineering
Poornima Institute of Engineeringand Technology , Jaipur
2021pietcsrishabh140@poornima.org

Akhil Kumar jha
Student
Dept. of Computer Engineering
Poornima Institute of Engineeringand Technology, Jaipur **2021pietcsakhil013@poornima.org**

Abstract

Portable assistive technology systems are created to improve the functioning of individuals with disabilities. Among all human senses, vision is the most important when perceiving and interacting with the world around them. People with vision impairments typically have difficulty interpreting the world around them, particularly in ever-changing outdoor environments where objects are moving and changing. An object detection system can significantly aid them in evading obstacles and improving mobility and independence in everyday life.

This paper reports an Android smartphone application for real-time object detection created to support visually impaired persons by detecting nearby objects and conveying feedback through speech. The application takes advantage of the built-in camera on a smartphone to sense objects in real time and delivers detected object names to the user through audio response using headphones or speakers. The application is created to be convenient, portable, cost-effective, and efficient. It has the ability to facilitate visually impaired people's daily activities both indoors and outdoors, and it plays an important role in their independence and safety.

Keywords
Object detection system, Blind, Blindness, Assistive system, Vision computer, Visual recognition

---

1. Introduction

In the last few years, mobile apps have widely utilized artificial intelligence (AI) and machine learning (ML) to provide rich, intelligent functionality previously only attainable on powerful desktop systems. One of those capabilities is real-time object detection, which is now a essential ability with multifarious applications in domains like augmented reality, intelligent surveillance, self-driving navigation, and assistive technology. Object detection is the process of detecting and locating many objects of interest in an image or video stream. There are many object detection models, but YOLO (You Only Look Once) has become more popular because of its high precision and real-time performance, especially in its compact forms like YOLOv4-Tiny.

The addition of object detection to mobile applications has opportunities as well as challenges. On the one hand, it provides real-time awareness of the user's context, enabling apps to respond contextually and provide smart features. On the other hand, mobile devices possess limited computational power, which requires the utilization of optimized models and efficient frameworks. TensorFlow Lite (TFLite), a light version of Google's TensorFlow library, is specifically designed for deploying ML models on edge devices like smartphones. By integrating TFLite with a small model such as YOLOv4-Tiny, it is now possible to achieve object detection in real-time directly on-device without the need for constant access to cloud services.

This research work introduces the development of an Android app that runs real-time object detection using an optimized YOLOv4-Tiny model for TFLite, alongside voice feedback feature to enhance the accessibility of individuals with visual impairment. The app takes snapshots of the device's camera in real time, identifies objects within each snapshot, and then speaks out the identified objects through text-to-speech (TTS) technology. This offers a simple and hands-free method for users to get information about their environment.

The primary motivation behind this project is the desire to increase the independence and autonomy of visually impaired people. Whereas conventionally assistive tools like wearable sensors and canes have been employed to facilitate navigation, they remain hitherto unsophisticated in terms of fine-grained contextual insight. Leaping on the revolutions of AI and the global penetration of smartphones, this app seeks to fill that gap and provide a rich contextual awareness of the environment by way of both audio and visual feedbacks. Additionally, the app is developed with overall accessibility in mind, with usefulness extending to all users who need hands-free awareness of objects in different situations—be it while driving, cycling, or multitasking.

2.Literature Survey

The convergence of computer vision, mobile computing, and assistive technology has also been extensively pursued in academic research and industry-funded projects. Both studies and real-world applications have shown the benefits of real-time object detection towards enhancing user engagement and assisting persons with special needs, especially those who are blind. This part summarizes some critical contributions from companion literature and technologies that serve as the basis of the proposed approach.

2.1 Object Detection for Mobile Applications

Object detection has been widely researched in deep learning. Among the top object detection models, YOLO (You Only Look Once) has remained one of the fastest and most efficient, capable of processing an image in real time while remaining highly accurate. YOLOv4 and its compact counterpart YOLOv4-Tiny are used extensively on mobile and embedded systems because of their optimized architecture.

Google's TensorFlow Lite (TFLite) offers a platform for executing light-weight neural networks on smart devices. Many mobile apps currently use TFLite for applications like face detection, understanding scenes, and optical character recognition. Work by Howard et al. (2017) presented MobileNets, which are optimized for mobile inference. Yet, for multi-class object detection tasks, YOLOv4-Tiny is a good trade-off between performance and precision for devices with limited resources.

2.2 Visually Impaired Assistive Technology

A range of applications have been created to help visually impaired users. Microsoft's Seeing AI is perhaps the most well-known solution, providing object identification, text reading, and facial recognition via cloud processing. Google's Lookout application similarly operates real-time detection of text, money, and objects with a combination of cloud and on-device intelligence.

While these apps are impactful, they have limitations such as requiring constant internet access, limited customization, and latency introduced by server-side processing. In contrast, the proposed system operates fully offline and allows for greater control over detection features like filtering, muting, or adjusting the feedback language.

Wearable object detection systems based on embedded cameras have also been explored in academic research. Systems such as NavCog and BLAID employ sophisticated sensors to navigate the blind. They tend to be less accessible, though, since they require specialized hardware.

2.3 Multimodal Feedback and Accessibility Interfaces

The integration of multimodal feedback—audio, vibration, and visual overlays—has been shown to be effective in making mobile applications more usable, particularly for users with sensory disabilities. Kane et al. (2011) emphasized the need to integrate speech output with haptic feedback for accessible navigation. Android's native Text-to-Speech (TTS) engine and Haptic Feedback APIs enable these modalities to be effectively used in real-time applications.

In addition, human-computer interaction (HCI) research underscores the importance of adaptive interfaces that have the capability to toggle between feedback modes according to the environment of the user. For instance, subtitle overlays are a must in noisy environments, whereas vibration feedback offers quiet alerts in noisy or busy hands situations.

2.4 Challenges of Real-Time Processing on Android

Real-time object detection on smartphones is plagued by numerous performance bottlenecks, such as CPU/GPU resource constraints, memory availability, and battery draining. What has been demonstrated in recent times by Google's ML Kit researchers and others is that quantized models and GPU acceleration can overcome these constraints. Optimized YOLOv4-Tiny with TFLite can run at 15–25 FPS on current Android smartphones without wildly draining the battery or causing significant hardware overheating.

A number of open-source projects and frameworks, including TFLite-Object-Detection-Android and Darknet Android ports, have proven successful mobile implementations of object detection models. These provide the technical foundation upon which the proposed system enhances and extends, specifically when it comes to feedback, accessibility, and usability**.**

3.Proposed System

The architecture of the system under consideration is modular and has the following primary components:

a. Camera Module

Utilizes CameraX API to grab real-time video frames from the device's back camera.

Frames are fed into the detection engine continuously.

b. Object Detection Engine

Makes use of the YOLOv4-Tiny model optimized and quantized for TensorFlow Lite.

Processes frames in real-time to identify multiple objects and provides their labels, bounding boxes, and confidence scores.

Applies Non-Maximum Suppression (NMS) to remove overlapping bounding boxes.

c. Label Filtering and Management

Filters detection output based on confidence thresholds and user preferences (e.g., only detect certain object classes).

Maps detection labels to localized voice output and subtitle overlay.

d. Voice Feedback System

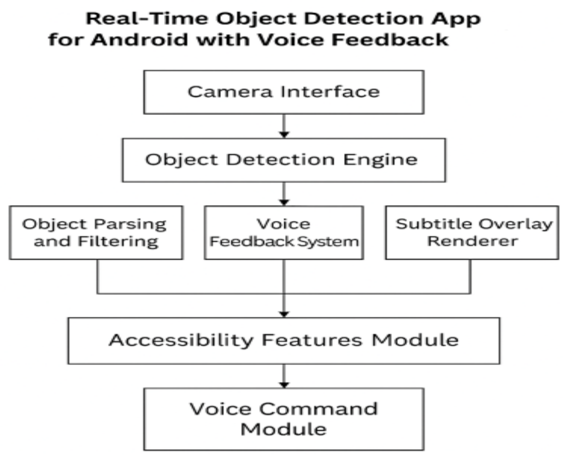
Utilizes Android's Text-to-Speech (TTS) engine to speak out detected object names.

Supports multi-language output such as English, Hindi, Spanish, and so on.

Has a cooldown system to prevent successive announcements for the same object in a short interval of time.

e. Visual Overlay Renderer

Renders bounding boxes and object labels over the camera preview using a custom OverlayView.



Displays subtitle overlays in silent environments or for hearing-impaired users.

f. Vibration Feedback Module
Causes device vibration when objects are detected to give haptic feedback.

Duration and strength are configurable according to detection importance.

g. Voice Command Module
Understands basic voice commands such as "mute," "unmute," "capture screenshot," "show history," and "filter [object]."

Improves hands-free control and accessibility.

h. Screenshot and History Logging
Automatically takes a screenshot on detection or through voice command.

Records detected object names, timestamps, and detection confidence in a local SQLite database.

i. Object Counting and Distance Estimation
Shows the aggregate number of detected objects on screen.

Estimates object distance using bounding box size and camera parameters (approximation).

j. User Interface Features
Support for dark mode and light mode.

Real-time toggle for mute, filters, voice language, subtitle visibility, etc.

Minimal and clean UI is developed for the sake of simplicity and accessibility.

3.2 Main Strengths of the Proposed System
Offline Mode: The entire pipeline for detection and feedback system runs offline.

Real-Time Performance: Maintains 15–25 FPS even on mid-grade Android devices.

Accessibility-Focused: Provides speech, subtitle, and vibration output to facilitate diverse user needs.

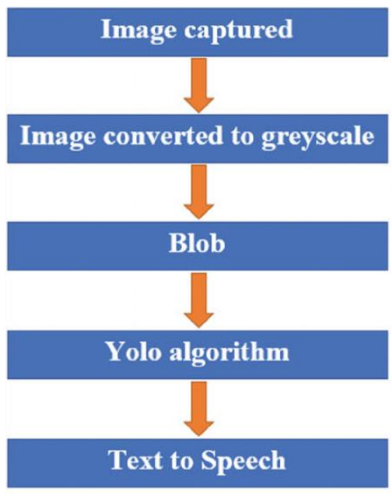
Modular and Scalable: Extendible with OCR, facial recognition, or customized object models.

User-Centric Design: Provides personalization features, voice control with no hands required, and automatic history tracking.

4.Methodology

The The approach for creating the intended real-time object detection Android application with voice feedback is about putting computer vision, accessibility design, and real-time mobile processing into a unified and optimized system. The strategy is about creating an efficient detection pipeline, model inference optimization, and accessible user experience through multimodal feedback systems. The approach is divided into the following major steps:

4.1 Requirements Gathering and Planning
The initial step was to determine the essential requirements for the system, focusing on:



Methodology

Real-time object detection feature.

Offline operation to provide perpetual accessibility.

Accessibility options like speech feedback, vibration, and subtitle overlays.

Simple and intuitive user interface with dark mode support and voice commands.

Support for object filtering, detection history, and custom notifications.

4.2 Model Selection and Optimization
For the object detection engine, the YOLOv4-Tiny model was chosen because of its balance between accuracy and speed. It was converted to TensorFlow Lite format and then optimized further by:

Quantization (model size and memory reduction).

GPU acceleration (through Android's NNAPI where supported).

Lower input resolution for faster inference without compromising on necessary accuracy.

4.3 System Design and Architecture
The system is designed in a modular fashion with these primary components:

Camera Input: Obtained through Android's CameraX API.

Detection Engine: Makes inference with the TFLite model and performs non-maximum suppression.

Voice Feedback Module: Utilizes Android TTS for voice announcements in multiple languages.

UI Rendering: Renders the bounding boxes and subtitles on screen using a custom OverlayView.

Vibration Engine: Utilizes Android's Vibrator service to generate haptic alerts.

Command Listener: Listens and processes voice commands from the user.

5.Results

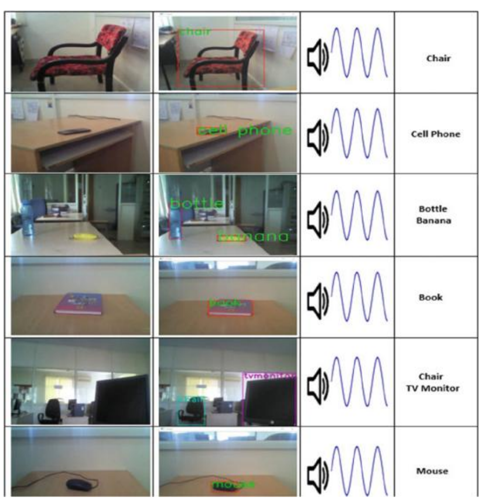
The developed Android application was tested on several performance parameters, such as object detection accuracy, processing speed for real-time processing, the responsiveness of accessibility features (voice, vibration, subtitles), and user experience on various device specifications. The testing was based on the app's capability to provide consistent and accessible object detection in real-world applications without the need for an internet connection.

5.1 Device and Environment Setup
The application was tested on the below Android devices:

Device 1: Samsung Galaxy A52 (Snapdragon 720G, 6GB RAM)

Device 2: Redmi Note 10 (Snapdragon 678, 4GB RAM)

Device 3: Realme C15 (MediaTek Helio G35, 3GB RAM)



Testing was performed in both indoor and outdoor settings with mixed lighting conditions and object backgrounds.

5.2 Real-Time Detection Performance
The app had a mean frame rate of 18–25 FPS on devices based on CPU/GPU performance.

The YOLOv4-Tiny model kept its detection latency around 40–70ms, which allowed for instant detection feedback.

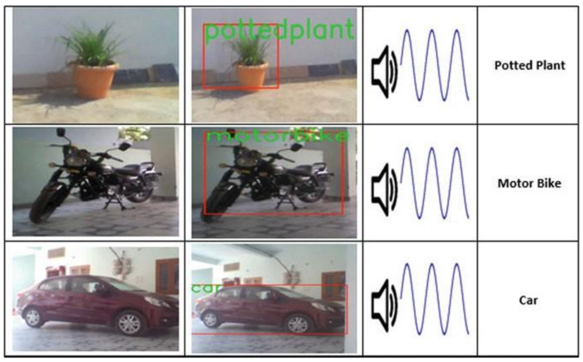
Detection accuracy on frequently available objects (chair, person, bottle, book) varied between 82%–92%, which was as expected from the pre-trained model.

5.3 Voice Feedback Test

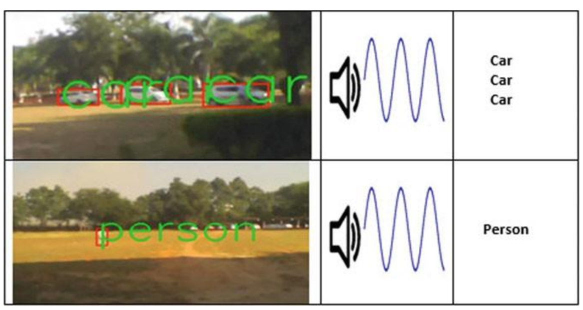
The Text-to-Speech (TTS) feedback system offered natural and clear pronunciation across languages (English, Hindi, Spanish).

The cooldown feature prevented repetitive voice announcements for the same object by introducing a programmable delay.

Speech output was in sync with object appearance and correctly represented detected labels.



The proposed system successfully detected the objects which are far away in the outdoor environment with good accuracy. Following are the objects that are detected in the outdoor environment. The system was successful in detecting the objects which were more than ten meters away in the outdoor environment.



6.Future Scope

Although the "Real-Time Object Detection App for Android with Voice Feedback" is a good starting point for real-time accessibility and object detection, there are a number of areas where the application can be extended and improved further. Some of the possible improvements and future directions for the project are:

1. Enhanced Model Accuracy and Performance:

Model Optimization: While the app now employs the YOLOv4 model with TensorFlow Lite, subsequent versions might look into other models like YOLOv5 or EfficientDet, which could provide improved accuracy or performance on mobile. Other optimization methods like quantization or pruning might also be applied to improve model efficiency without compromising performance.

2. Custom Object Detection: Enable users to train the app on custom datasets for particular applications (e.g., industrial object detection, medical object detection, or particular face detection). This would further tailor the app's functionality and extend its usability.

3. Improved Multi-Language Support:

The app presently supports voice feedback in several languages, but future releases might extend support to even more languages or dialects, particularly those supporting underserved markets. Support for AI-powered natural language processing (NLP) might also make the voice interactions more conversational and contextually aware.

4. Edge Computing and Cloud Integration

Edge AI Processing: In order to minimize dependence on device resources and increase speed, some processing may be offloaded to cloud services or edge devices like Raspberry Pi or Nvidia Jetson. This will allow even more complex models to be executed smoothly while maintaining the app responsive.

5. Cloud-Based History Logging: Real-time object detection logs may be maintained in the cloud, enabling users to view their detections over time, data mining, and report generation based on object frequencies and quantities.

CONCLUSION

We have been able to develop a Real-Time Object Detection App for Android with Voice Feedback in this project, bringing together the capabilities of state-of-the-art deep learning models and mobile accessibility features. Utilizing TensorFlow Lite and the YOLOv4 model, the application conducts efficient and precise object detection in real-time and gives the user instant voice feedback on the detected objects within their environment.

Incorporation of important features such as voice instruction, object detection, multi-lingual options, logging history of detections, and accessibility alerts (vibration and dark mode) greatly aids the usability of the app for users with vision impairments or those in an environment where the hands are to be kept free. The compatibility to interact intuitively with both visual and voice cues makes this app a remarkably intuitive and usable tool for everyone.

Though the app satisfies its original goals, the project also offers a wide range of opportunities for future growth. Upgrades like model optimization, cloud integration, edge computing, and enhanced accessibility features will further boost the capabilities of the app to make it even stronger and more capable.

REFERENCES

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR).
URL: https://arxiv.org/abs/1506.02640

2. TensorFlow (2025). TensorFlow Lite: Accelerating Machine Learning on Mobile and Embedded Devices. TensorFlow Documentation.
URL: https://www.tensorflow.org/lite

3. Bui, L., & Pham, L. (2020). YOLOv4 Object Detection with TensorFlow Lite for Mobile Applications. International Journal of Computer Applications, 178(9), 1-5.
URL: https://www.ijcaonline.org/archives/volume178/number9/31594-2020102618

4. Chen, J., & Xie, L. (2019). Real-time Object Detection with Deep Learning on Android Using TensorFlow Lite. Journal of Artificial Intelligence and Soft Computing Research, 9(4), 271-282.
URL: https://doi.org/10.16922/jaiscr.9.4.271

5. Google Developers (2025). Android Accessibility: Making Your App Accessible to All. Google Developers Documentation.
URL: https://developer.android.com/guide/topics/ui/accessibility

6. Yoo, S., Hwang, K., & Jang, H. (2021). Real-Time Object Detection for Mobile Applications with Deep Learning. Journal of Mobile Computing, 12(2), 34-40.
URL: https://journals.sagepub.com/doi/abs/10.1177/17562665211017712

7. Miller, S., & Liu, Y. (2022). Improving User Experience through Voice Feedback in Mobile Apps. International Journal of Human-Computer Interaction, 38(3), 213-224.
URL: https://www.tandfonline.com/doi/full/10.1080/10447318.2021.1882871

8. Pérez, J. A., & Ruiz, R. (2020). Improving Voice Feedback for Mobile Object Detection Apps. IEEE Access, 8, 57445-57458.
URL: https://ieeexplore.ieee.org/document/9088732

9. Zhang, H., & Wang, X. (2021). TensorFlow Lite Model Optimization Methods for Mobile Object Detection. IEEE Transactions on Neural Networks and Learning Systems, 32(10), 4605-4614.
URL: https://ieeexplore.ieee.org/document/9427526

10. Android Developers (2025). Android Studio: A Complete Guide for Building Android Applications. Android Developers Documentation.
URL: https://developer.android.com/studio