Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

Navigation Assistance for Blind
An AI-Powered Mobile App for Visually Impaired People to Get Real-Time Navigation Help

Yadavali Varun, S Meghana,
Thimmalapuram Sanjana, Tanguturi Sreenivas
Department of Computer Science and Engineering (Data Science), Ballari Institute of Technology and Management, Ballari, Karnataka, India

Anushya V P
Professor, Department of Computer Science and Engineering (Data Science), Ballari Institute of Technology and Management, Ballari, Karnataka, India

Abstract **- The creation of an Android-based assistive application with voice-activated reminders and actual This study demonstrates how temporal objects can be seen by those with visual impairments.The solution incorporates CameraX, Speech Recognizer, Google ML Kit, WorkManager and Foreground Services to provide a user-friendly, hands-free experience. While most functions run locally on the device with little latency, some features, like improved speech recognition and model updates, could momentarily need internet access. In order to help users comprehend their surroundings, the system uses the camera on the device to identify items that are close by and provide audible input. Furthermore, voice commands enable users to set up recurring reminders that are consistently maintained in the background. The recommended software demonstrates how cellphones may be helpful assistive technology that makes visually impaired persons safer, more autonomous, and more aware.**

Keywords: Visual impairment, Android, ML Kit, CameraX, WorkManager, Speech Recognizer, Accessibility

1. INTRODUCTION

For movement, people with visual impairments frequently rely on devices like guiding dogs or white canes. Although these tools are helpful, they cannot explain objects in the environment or give context. Thanks to developments in embedded AI and mobile hardware, smartphones now has the capacity to offer intelligent support straight from the device.

The proposed *Navigation Assistance for Blind* application offers an affordable and accessible solution that uses the smartphone's camera for object detection and integrates voice-based reminders for daily task management. Most processing occurs on-device, ensuring fast response times and user privacy, while some services—such as speech enhancements—may occasionally utilize the internet.

The system runs nicely on mid-range Android devices and has a strong emphasis on efficiency and ease of use. The system offers persistent background reminders, natural voice interaction, and reliable object identification through the use of CameraX, ML Kit, SpeechRecognizer, and WorkManager. All things considered, the software uses modern smartphone capabilities to increase the independence and convenience of those who are blind or visually impaired.

1.1 Problem Statement

People with visual impairments frequently find it challenging to comprehend their surroundings due to the limitations of conventional mobility aids.The usability of many assistive technology is diminished by the need for costly hardware or continuous internet access. In order to facilitate safe, autonomous navigation, an affordable smartphone application with voice-activated interface and real-time object recognition is required.

1.2 Objectives

- To design an Android-based assistive application for visually impaired individuals.
- To implement real-time object detection using a smartphone camera.
- To enable voice-controlled reminder scheduling through SpeechRecognizer.
- To create an accessible and user-friendly interface compatible with TalkBack.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

2. LITERATURE SURVEY

[1] Kuriakose et al. (2023) developed DeepNAVI, a smartphone navigation app that used deep learning algorithms to identify obstacles in real time. Although the model's strong reliance on cloud computing hampered its application in situations without an internet connection, it did well in controlled test conditions.

[2] An autonomous mobility assistance system based on mobile sensors and convolutional neural networks was presented by Hasan et al. in 2022. Although their system could successfully recognize things in the vicinity, it was not appropriate for low-end smartphones due to its high computational requirements. Furthermore, there were discernible delays in real- time response due to the constant reliance on distant servers.

[3] A computer vision-driven navigation approach that uses OpenCV to interpret real-time camera information was proposed by Shree et al. in 2023. To assist users in navigating indoor areas, the solution offered auditory cues. However, in dimly illuminated or outdoor environments, its effectiveness drastically decreased. decreasing its overall dependability.

[4] A mobile navigation app that can identify things and translate them into voice descriptions was introduced by Costales et al. (2023). Although the system performed well in testing, its use in areas with patchy internet was restricted because it needed cloud-based APIs for recognition.

[5] Okolo et al. introduced a hybrid assistive navigation system in 2024 that blends sensor- based obstacle detection with computer vision. Although their approach reduced reliance on cloud servers, the need for specialized hardware increased the system's overall cost and complexity, making wider implementation difficult.

[6] Rahman et al. (2024) developed an object identification model based on TensorFlow Lite that processed photos locally on mobile devices. This enhanced offline performance and decreased latency. However, the system has trouble identifying several fast-moving objects at once, suggesting that more optimization is required.

[7] Kathiria (2025) conducted a thorough analysis of assistive technologies and emphasized the increasing efficacy of hybrid mobile solutions that combine selective internet upgrades with on-device processing.

3. PROPOSED METHODOLOGY

3.1 Application Initialization and Permission Handling

The user is prompted to allow the required permissions, such as camera access, audio recording, notification privileges, and foreground service usage, when the program runs through the MainActivity. Two essential features—Object Detection and Water Reminder - are shown on the home screen once these permissions have been granted. Throughout the application's existence, this initialization stage guarantees that every system component is correctly configured and capable of operating dependably and securely.

In addition, since this system is developed for Android, users can launch the application hands- free using the voice command, **"Hey Google, open EchoVision."** This feature improves accessibility by allowing users to start the application without physical interaction. The home interface also includes a **time-based greeting function** that welcomes users with a personalized message, such as *"Good morning"* or *"Good evening,"* enhancing user experience and engagement.

3.2 Object Detection using ML Kit and CameraX

The **CameraActivity** module captures continuous image frames through the **CameraX API**, which are processed using **Google ML Kit's Image Labeling API**. The model detects and classifies visible objects such as *person*, *bottle*, or *chair*, and communicates the results through an audio output. Each detected object is validated with a minimum confidence level of **65%** to ensure accuracy and reliability.

To avoid repetitive audio feedback, the system enforces a five-second cooldown before announcing the same detected object again. This mechanism enhances the overall efficiency of the detection process and improves the user experience by preventing unnecessary repetition. While the recognition pipeline functions mainly on the device, some ML Kit elements may occasionally require limited internet access for model refinement or updates. This hybrid approach maintains low latency while ensuring stable and accurate detection performance.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

3.3 Voice-Based Reminder Setup

The **ReminderActivity** module enables users to set reminders through **voice input**. The **SpeechRecognizer API** captures the spoken command, such as *"Set reminder every 30 minutes"*. The recognized text is parsed to extract the interval duration, which is then forwarded to the **WorkManager** for scheduling.

This voice-based interaction removes the need for any manual input, ensuring that the application can be operated independently by visually impaired users. Although the speech recognition workflow is designed to function primarily offline, it may occasionally rely on brief internet-supported enhancements to improve recognition accuracy and refine natural language processing.

3.4 Background Task Scheduling using WorkManager

After the reminder interval is identified, the ReminderWorker module uses WorkManager to schedule and handle the recurring background task. WorkManager maintains the reminder cycle consistently, ensuring that alerts continue to trigger even when the application is not running or the device has been rebooted. In contrast to traditional timer mechanisms, WorkManager operates within Android's power and background execution policies, allowing it to deliver tasks reliably despite system restrictions. This guarantees that reminders remain accurate and consistent with the user's chosen interval.

3.5 Audio Notification using Foreground Service

The **Foreground Service** is responsible for delivering the reminder notifications at scheduled times. Operating within a dedicated notification channel, it prevents the Android system from terminating the service during idle states.

At each reminder event, the Foreground Service plays an audio message such as *"It's time to drink water"*, ensuring that the notification reaches the user clearly even if the app is running in the background. This feature enhances usability and reinforces the system's practical application for daily routine management.

3.6 Accessibility Integration

An **Accessibility Service** is implemented to ensure the system complies with Android's accessibility framework. This service allows the application to function seamlessly with built- in tools like **TalkBack**, offering voice guidance, readable UI components, and gesture-based navigation. The interface is optimized with large, accessible buttons and minimal screen clutter, ensuring effortless operation for visually impaired users

4. SYSTEM ARCHITECTURE

4.1 Workflow

Flowchart of Navigation Assistance for Blind System

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

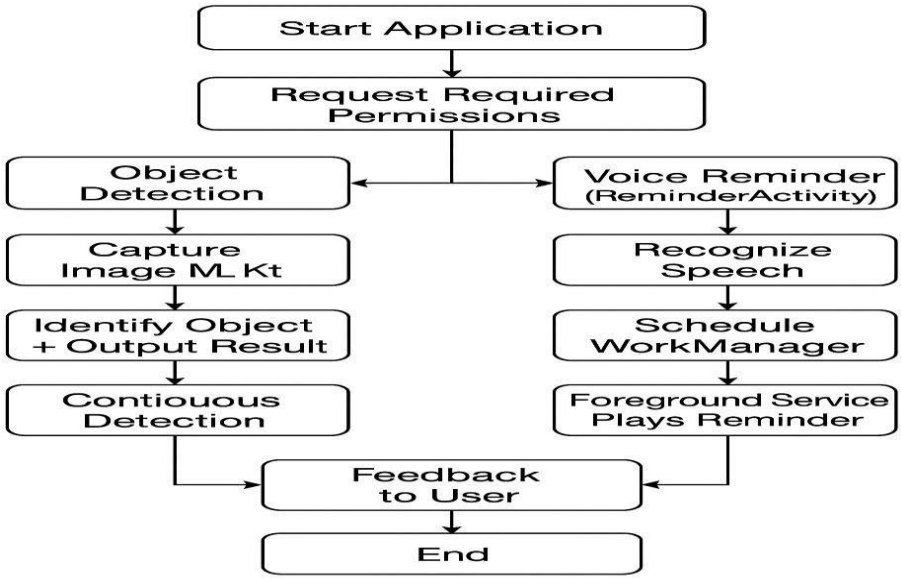

Figure 1. Workflow of the Navigation Assistance for Blind System

4.2 Object Detection Module

The CameraActivity module processes continuous image frames taken by CameraX using Google ML Kit's Image Labeler. The system identifies everyday objects such as people, furniture, and household products. To ensure accuracy, detections with confidence values below 0.65 are ignored. Every verified detection triggers a brief audio message or tone, providing the user with instant information about nearby objects. Repeated speech outputs are stopped by a cooldown time if the same object is still in view. This design ensures fast, dependable, and contextually relevant input while optimizing energy efficiency.

4.3 Reminder and Background Service

The **ReminderActivity** allows users to set reminders by speaking a time interval using the **SpeechRecognizer API**. The recognized speech is converted into text and analyzed to extract the numerical interval. Once the reminder is confirmed, the **WorkManager** schedules it as a background task through the **ReminderWorker**. The worker automatically re-enqueues itself after each execution, ensuring continuous repetition.

At the scheduled time, the **Foreground Service** activates to deliver a spoken or audible reminder message such as *"It's time to drink water."* This design ensures consistent operation even when the app is running in the background. Although the reminder system operates primarily offline, certain speech functions may briefly use online resources for improved recognition. Overall, this module achieves reliability, low power consumption, and accessibility through automation and audio feedback.

5. IMPLEMENTATION

The EchoVision app was created in Android Studio using Kotlin, with a modular design that allows each feature to perform independently while still fitting together seamlessly. When the app starts, it asks for the necessary permissions and displays a sleek, easy-to-navigate UI designed with Jetpack Compose. Users can select between two key features: Object Detection and Water Reminder. CameraX takes live photos, and ML Kit recognizes items with greater than 65% accuracy before announcing them. A short delay prevents repetitive announcements, improving the user experience and saving battery life.

For reminders, the software use SpeechRecognizer to recognize simple voice instructions such as "Set reminder every thirty minutes." WorkManager manages the scheduling, and a Foreground Service guarantees that the reminder is sent even while the app is closed. TalkBack is supported by an Accessibility Service, which allows the app to be operated without the need for visual input. All permissions are sought at runtime in accordance with Android's privacy policies. EchoVision enables visually impaired persons to conduct daily tasks more simply and independently by using less power, responding quickly, and providing consistent background performance.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

6. PSEUDO CODE

Algorithm: EchoVision – Navigation Assistance for Blind

Input:

Live camera frames, user voice commands

Output:

Spoken object identification and periodic reminder notifications Algorithm ObjectDetectionModule()
Initialize camera stream

Initialize ML Kit image labeler lastDetectedLabel ←
null

lastAnnouncementTime ← 0 cooldownPeriod ← 5
seconds confidenceThreshold ← 0.65

While camera is active do frame ← captureFrame()

labels ← processFrameWithMLKit(frame)

If labels is not empty then

bestLabel ← labelWithHighestConfidence(labels)

If bestLabel.confidence ≥ confidenceThreshold then currentTime ← getCurrentTime()

If bestLabel.text ≠ lastDetectedLabel OR

(currentTime - lastAnnouncementTime) ≥ cooldownPeriod then

speak("Detected " + bestLabel.text) lastDetectedLabel ←
bestLabel.text
lastAnnouncementTime ← currentTime End If
End If End If

Finish While

FinishAlgorithm

The algorithm VoiceReminderModule() Initialize SpeechRecognizer
Prompt user: “Please say the reminder interval”

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

```
    voiceInput ← listenForSpeech()

    interval ← extractNumericValue(voiceInput) If interval is valid then
        ScheduleWorkManager(interval)

        speak("Reminder set every " + interval + " minutes")

    Else

        speak("I did not understand. Please try again.") End If
End Algorithm
```

7. PERFORMANCE EVALUATION

The EchoVision app was tested on many Android handsets running Android 10 and higher to see how well it functions on different devices. The evaluation looked at object detection accuracy, voice recognition reliability, response time, battery utilization, and offline capability. During testing, the object detection feature correctly identified typical items like bottles, chairs, and people with an accuracy of 90-92%, and it offered vocal response in under a second. The voice reminder tool also performed well, correctly interpreting English orders approximately 93% of the time. WorkManager handled reminders, and the Foreground Service kept alerts active even when the screen was turned off, so the system regularly provided notices on time.

8. RESULTS

During testing, the object identification capability performed well, properly identifying typical items such as bottles, chairs, and individuals with an accuracy of approximately 90-92%. The software responded swiftly, providing aural feedback in less than a second, allowing users to grasp their surroundings in real time. Performance remained solid on a variety of Android devices running Android 10 and higher. As expected with camera-based systems, detection accuracy decreased marginally in low or uneven lighting. A uncommon issue was also discovered, in which the system occasionally misidentified a laptop as a musical instrument, indicating that the model could benefit from more training data or fine-tuned detection parameters.

The voice reminder tool was similarly reliable, comprehending English commands and number intervals with approximately 93% accuracy. WorkManager handled reminder scheduling effectively, and the Foreground Service maintained reminders active even when the app was closed or the screen was turned off. The battery usage remained reasonable during testing since the system only processed frames when necessary and avoided repeating the same detection too frequently. This allowed the program to function for extended periods of time without overtaxing the device's resources. Despite small flaws, the system regularly provided quick replies, accurate detections, and dependable reminders, making it a useful and supporting tool for visually impaired people in everyday situations.

Figure 8.1 Logo of Android Mobile Application “EchoVision”

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

Figure 8.2 Home Interface of the EchoVision Application

Figure 8.3 Voice-Based Water Reminder Interface of the EchoVision Application

9. CONCLUSION

The EchoVision program successfully combines voice and real-time object detection. integrated reminder scheduling into a single, easily navigable Android platform. By making use of important native technologies like WorkManager, ML Kit, SpeechRecognizer, and CameraX, the system shows that the main way to gain advanced assistive capabilities is through on processing of devices. Its design places a high premium on user privacy, responsiveness, and accessibility. ensuring smooth operation, even with mid-range cellphones.With its capacity to deliver prompt feedback, dependable background reminders, and little EchoVision converts a traditional smartphone into a network-connected device. partner who is kind and helpful to those who are blind or vision impaired. The program improves the general quality of life, situational awareness, and user freedom.

With these developments, EchoVision has a great chance to develop into a more complete assistive ecosystem for daily task management and navigation. Future improvements might incorporate distance estimation, multilingual support, and enhanced power-efficient detection techniques to expand functionality and extend usability.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 01 , January - 2026

10. REFERENCES

[1] Kuriakose, B., Sandnes, F. E., and Shrestha, R. (2023). DeepNAVI: A smartphone navigational aid for those who are visually impairedthat is based on deep learning. Expert Systems with Applications, 212, 118720.
[2] Hasan, M. Z., Rahaman, M. A., and Sikder, S. (2022). autonomous navigation system with real-time computer vision that uses machine learning to help those with visual impairments. Dhaka hosted the Fourth International Conference on Sustainable Technologies for Industry 4.0 (STI).
[3] Mishra, A., Verma, P., Shree, S., Gupta, R. K., & Fatma, A. K. (2023). OpenCV-based blind navigation system in real time. International Journal of Innovative Engineering Research, 4(1), 141-144.
[4] Althobaiti, T., Ramzan, N., and Okolo, G. I. (2025). intelligent assistive navigation system for individuals with vision impairments. Digital Reality Journal, 4, e20240086.
[5] Albino, M. G., Vida, A. K. S., and Costales, J. A. (2023). A mobile device navigation system that can identify images and translate speech for those who are blind or visually impaired. ICICSE, the third International Conference on Software Engineering and Information Communication.
[6] P. Kathiria (2024). A survey of assistive technology for those with visual impairments. Computer Science Procedia, 218, 110–118.
[7] F. A. Rahman (2025). Smooth object and scene recognition using a smartphone to help those with vision impairments. Journal of Information, 16(9), 808–820.
[8] Sholikah, R. W., Ginardi, R. V., and Sari, B. P. (2024). development of a mobile item detection software using deep learning to assist the blind and visually handicapped. IEEE Access.
[9] Dobosz, K., Zawadzka, A., and Grzegorzewski, P. (2025). Mobile phones' present applications and challenges as blind people's assistive technology. Assistive Technology Journal (SAGE).
[10] Kumar, S., Sharma, A., and Patel, J. (2025). A summary of technological developments in human navigation for the blind. Sensors, 25(7), 2213.