MASTER'S DEGREE IN WEB TECHNOLOGY AND SYSTEMS ENGINEERING

DISSERTATION PROJECT

ASSISTIVE MOBILE APPLICATION FOR VISUALLY IMPAIRED BASED ON REAL TIME OBJECT RECOGNITION USING MACHINE LEARNING WITH VOICE FEEDBACK

RABI BEN RHOUMA

SUPERVISOR : FIRMINO SILVA

DISSERTATION PROJECT

VILA NOVA DE GAIA
11/2025

Master's Dissertation submitted for partial satisfaction of the requirements for the Master's degree carried out under the supervision of Professor Firmino Silva presented to ISLA - Polytechnic Institute of Management and Technology of Vila Nova de Gaia to obtain the Master's degree in WEB TECHNOLOGY AND SYSTEMS ENGINEERING, in accordance with the Order n. º 9371/2020.

INSTITUTO POLITÉCNICO DE GESTÃO E TECNOLOGIA

Assistive Mobile Application for Visually Impaired Based on Real Time Object Recognition using Machine Learning with Voice Feedback

Rabi Ben Rhouma

Aprovado em 19.12.2025

Composição do Júri

Presidente
Prof. Doutor Jorge Pereira Duque

Arguente
Prof.ª Doutora Célia Talma Gonçalves

Orientador
Prof. Doutor Firmino Oliveira da Silva

Vila Nova de Gaia
2025

Abstract

Visual impairment significantly restricts individuals' autonomy, mobility, and capacity to interact with their surroundings. According to the World Health Organization, more than 2.2 billion people live with vision impairment, many of whom face daily challenges in navigating unfamiliar environments, avoiding obstacles, and identifying essential objects [1]. Recent advancements in artificial intelligence (AI) and computer vision offer promising opportunities to develop assistive technologies capable of addressing these challenges through real-time environmental understanding.

This project presents the development of a mobile application designed to assist visually impaired individuals through real-time object detection combined with audio feedback. Three different approaches were explored. The first two approaches used custom deep-learning models based on YOLOv5, trained on datasets of varying size and complexity. While these models demonstrated good accuracy, deployment on mobile devices proved challenging due to computational constraints and limited class coverage. The final approach leveraged the Google Cloud Vision API to off-load inference to the cloud, enabling broad object recognition without the need for local training or heavy computation.

The resulting system captures an image through the Android camera, processes it via a FastAPI backend, and returns a concise voice description of detected objects. User interviews guided the design of the interface, ensuring accessibility and simplicity for individuals with different levels of visual impairment.

The results demonstrate that cloud-based computer vision significantly improves recognition breadth, reliability, and real-time performance. However, trade-offs include dependence on network connectivity and third-party services. The final prototype shows strong potential as a low-cost, scalable assistive solution. Future work will explore hybrid systems combining offline detection for essential objects with cloud-based recognition for complex scenes.

Keywords: Object Recognition, Assistive Technology, Computer Vision, YOLO, Google Cloud Vision API, TensorFlow Lite, Visually Impaired Users.

Acknowledgments

I would like to begin by expressing my deepest gratitude to **my parents**, whose love, sacrifices, and unwavering support made this entire journey possible. They gave me the opportunity to pursue my studies in Portugal, supported me financially and emotionally, and believed in me even during the most challenging moments. Their encouragement has been the foundation of my success, and I dedicate this work to them with profound appreciation.

I would also like to sincerely thank my supervisor, **Professor Firmino Silva**, for his continuous guidance, constructive feedback, and valuable expertise throughout the development of this dissertation. His patience, availability, and commitment greatly contributed to the quality and direction of this work.

My heartfelt thanks go as well to the **visually impaired participants** who generously took part in the interviews. Their insights, experiences, and willingness to collaborate were essential in shaping a solution that truly responds to real user needs. Without their contribution, this project would not have achieved its intended purpose.

I am also grateful to my **friends and colleagues**, who offered constant motivation, advice, and companionship throughout this process. Their support helped me stay focused and overcome the difficult stages of the project.

Finally, I would like to thank **ISLA Gaia** for providing the academic environment, resources, and necessary support to complete this research.

To all of you, thank you for your trust, encouragement, and support.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Problem contextualization and motivation

At present visual impairment is one of the most common disabilities worldwide, affecting more than 2.2 billion people, according to the world report on vision of the World Health Organization (WHO) [1]. From these cases, at least 1 billion could have been prevented or have yet to be addressed due to lack of access to treatment [1]. Visual impairment, ranging from low vision to complete blindness, significantly hampers individuals' ability to navigate and interact with their environment. Everyday tasks such as walking in a crowded area, crossing streets, or identifying common objects become increasingly challenging for those with severe vision loss. This restriction on mobility and autonomy often leads to dependency on others for basic activities, affecting both mental well-being and overall quality of life [1].

To alleviate these issues, assistive technologies have emerged over the past decades, leveraging advancements in artificial intelligence (AI) and machine learning (ML). From rudimentary white canes to sophisticated smart glasses, these tools aim to enhance the situational awareness of visually impaired users by detecting obstacles, recognizing objects, and providing haptic or audio feedback [2]. Nevertheless, despite the surge in these technologies, many existing solutions still exhibit significant shortcomings when applied in real-world scenarios.

One notable shortcoming is the lack of real-time feedback, a crucial factor for visually impaired individuals who rely on constant and up-to-date information about their surroundings. For instance, the ability to quickly detect obstacles or identify traffic signals while crossing the street requires low-latency response times that many current assistive tools fail to deliver [3]. Moreover, another common limitation is the inability to adapt to different environments, such as outdoor settings with varying light conditions or indoor spaces cluttered with diverse objects [4].

Recent advancements in AI, particularly in the field of computer vision, offer promising solutions to these challenges. AI-driven object detection algorithms, such as YOLO (You Only Look Once), have demonstrated remarkable accuracy in recognizing a wide range of objects and actions within a scene [5]. More importantly, these models can be integrated into mobile devices using lightweight versions such as TensorFlow Lite, enabling real-time processing of visual data directly on smartphones [6]. The combination of real-time object recognition with immediate voice feedback provides a more holistic assistive experience, enhancing the independence and mobility of visually impaired users [7].

Mobile applications utilizing AI can identify obstacles, landmarks, traffic lights, and even specific behaviors like waving hands or riding a bike. By recognizing these elements, AI-powered solutions can significantly improve navigation and safety for visually impaired individuals, allowing them to receive audio instructions to safely navigate their environment. The ability to provide actionable information, such as "red light ahead" or "person approaching from the left," is a major leap forward compared to more passive, outdated technologies.

The motivation behind this project is driven by the urgent need to address the challenges faced by visually impaired individuals in maintaining autonomy and independence in daily life. Existing technologies, while useful, fall short in providing real- time, context-aware feedback that adapts to varying environments. The proposed mobile application aims to fill this gap by integrating AI-powered object recognition, real-time feedback, and a user-friendly interface that tailors its responses to the specific needs of visually impaired users.

The innovation lies in combining computer vision algorithms like YOLO with voice output systems, which together provide both detailed object recognition and immediate feedback. This hybrid approach enables users to perceive and react to dynamic environments such as busy streets, unfamiliar indoor settings, and even complex environments like shopping malls or transportation hubs. Through the proposed application, visually impaired individuals will gain increased autonomy, reducing their dependence on others and improving their quality of life.

The challenges of visual impairment call for robust technological interventions. This project is motivated by the potential of AI to revolutionize assistive technology, offering not just a tool but an interactive, real-time companion that can guide visually impaired users in their daily activities. By building on previous advances in computer vision and AI, this work seeks to develop a practical and reliable solution that empowers individuals to live more independently in a sighted world.

1.2 Objectives

1.2.1 Global objective :

The primary goal of this project is to improve the autonomy and quality of life for visually impaired individuals by enhancing their independence and mobility. By providing real-time object recognition and environmental feedback, the mobile application will empower users to navigate both indoor and outdoor environments with greater confidence and safety. The use of advanced AI techniques ensures that the app delivers a reliable and user-friendly experience, reducing the need for external assistance in daily activities and ultimately fostering a greater sense of independence.

1.2.2 Specific objectives :

To achieve the global objective of improving the autonomy and quality of life for visually impaired individuals, this project is divided into several key steps, each designed to address specific aspects of the mobile application's development and functionality. These specific objectives are critical for ensuring the effectiveness, accessibility, and real-world usability of the proposed solution. They focus on the technical development of the AI model, optimization for mobile performance, and user-centered design, all of which are essential for delivering a practical and impactful assistive tool. The specific objectives are as follows:

- Develop and Train an AI Model for Object Detection:

Train a custom object detection model using YOLOv5 to recognize and classify a variety of objects relevant to both indoor and outdoor environments (e.g., traffic lights, bus stops, furniture, personal items). This ensures the model can accurately identify essential objects in everyday scenarios.

- Optimize the AI Model for Mobile Devices:

Convert the trained model to TensorFlow Lite format for efficient, real-time performance on mobile devices, ensuring low latency and smooth user experience. This is crucial for maintaining real-time feedback, which is critical for visually impaired users.

- Design a User Interface with Voice Feedback:

Develop a mobile application with an intuitive user interface (UI) that provides real-time voice feedback to users, alerting them of objects and important elements in their surroundings. Voice feedback is essential for guiding users who cannot rely on visual information.

- Test the Application in Real-world Scenarios:

Conduct tests in various environments (indoor, outdoor, under different lighting conditions) to evaluate the model's accuracy, speed, and reliability in providing timely and useful feedback to visually impaired users. This ensures the application's effectiveness in diverse and dynamic real-life situations.

- Ensure Accessibility and User-friendliness:

Incorporate feedback from visually impaired users to ensure that the application is accessible, easy to use, and meets their specific needs in terms of navigation, object detection, and overall interaction with the environment. This guarantees that the app is designed with the end-user in mind.

- Deploy the Application for Android Devices:

Implement the application for Android devices using Kotlin and TensorFlow Lite, ensuring compatibility with common smartphones and low-resource devices for widespread use. This increases the accessibility of the app for a broader audience.

1.3 Expected Results

This project aims to deliver a practical, reliable, and user-centered mobile application that enhances the autonomy and quality of life for visually impaired individuals. The expected outcomes can be grouped into several categories, reflecting the technical achievements and the real-world impact of the application:

- High-Accuracy Object Detection Model:

A custom-trained object detection model using YOLOv5 will accurately identify a wide variety of relevant objects in both indoor and outdoor settings. This model will be capable of recognizing traffic lights, bus stops, pedestrians, furniture, personal items, and other environmental objects that are essential for safe and independent navigation.

- Real-time Performance on Mobile Devices:

The AI model will be optimized for mobile performance through conversion to TensorFlow Lite format, ensuring real-time object recognition and feedback. This guarantees that users receive low-latency responses, a critical feature for navigating fast-paced or dynamic environments, such as crossing streets or avoiding obstacles.

- Intuitive Voice Feedback System:

The mobile application will feature a user-friendly interface with real-time voice feedback, guiding visually impaired users by informing them of detected objects and environmental elements. This will enable users to make informed decisions quickly and efficiently, improving their mobility and confidence in various scenarios.

- Enhanced User Experience for the Visually Impaired:

The app will be designed to ensure accessibility, with a focus on ease of use for visually impaired individuals. Voice feedback, simple navigation, and responsive controls will ensure that the application provides a seamless and enjoyable user experience, meeting the specific needs of the target audience.

- Robust Performance Across Diverse Environments:

The application will be tested in various real-world environments, including different lighting conditions (day and night), indoor and outdoor settings, and cluttered spaces. The goal is to ensure that the model maintains high accuracy and reliability, regardless of environmental variability, providing consistent support for visually impaired users in their everyday activities.

- Positive Impact on User Independence and Quality of Life:

Ultimately, the mobile application will empower visually impaired individuals to navigate their environments with greater independence and confidence. By reducing reliance on others for tasks like object recognition and navigation, the app will enhance the user's autonomy, safety, and overall quality of life.

- Scalable and Accessible Solution:

The deployment of the application on Android devices will ensure that it is accessible to a broad audience, particularly in regions where smartphones are more prevalent. The app's compatibility with low-resource devices will also make it a viable solution for individuals in diverse socioeconomic settings.

This project is expected to result in an AI-driven, real-time assistive mobile application that will provide visually impaired individuals with the tools they need to navigate both familiar and unfamiliar environments safely and independently. Through the combination of advanced computer vision, voice feedback, and user-centered design, this application has the potential to significantly improve the daily lives of its users.

1.4 Methodology: Design Science Research (DSR)

This project will employ the Design Science Research (DSR) methodology, a well-established approach in information systems and engineering research that focuses on the creation and evaluation of innovative artifacts to solve identified problems [8], [9]. DSR is particularly suitable for this project as it provides a structured framework for

developing and assessing technological solutions, such as the proposed mobile application for visually impaired individuals.

1.5 Organization of the report

This report is organized as follows: Section 2 reviews the state of the art in image recognition, assistive applications, and existing solutions for visually impaired individuals. Section 3 details the methodology, design process, dataset preparation, and model development. Section 4 presents the results and evaluation of the first two approaches. Section 5 introduces the third approach using the Google Cloud Vision API, and Section 6 concludes the work by summarizing the findings and outlining future research directions.

2. State of the Art

2.1 Image Recognition Technologies

Several studies have explored the use of AI-based technologies for object recognition in assistive applications for visually impaired individuals. Object detection and image recognition algorithms like SSD (Single Shot MultiBox Detector), YOLO (You Only Look Once), and convolutional neural networks (CNNs) are the most commonly used [10], [11], [12], [13].

For instance, in [14], the authors developed an Android-based object recognition application using TensorFlow's object detection model, which leveraged the SSD algorithm for real-time and offline object detection. The system achieved a Mean Average Precision (mAP) of 74.3% for detected objects. However, the researchers encountered difficulties with integrating real-time voice feedback, which is crucial for assisting visually impaired users.

Another study by [15] focused on designing an assistive application of obstacle detection based on deep learning for visually impaired people using YOLOv3 with a Darknet-53 base network. Their mobile application works in real time to detect obstacle with high-speed detection and high accuracy. In addition, it generates audio output with the name of the obstacle detected in different languages, As shows the figure 1.

Figure 1: The output of the system prototype on the smartphone [15]

Further advancements in object recognition are seen in [16], where a CNN-based approach using OpenCV and the COCO dataset was applied to train YOLO. This method enabled the system to identify multiple objects in a single image, further improving the practical utility of assistive apps for visually impaired people.

2.2 Using AI for Visually Impaired People

AI's role in empowering visually impaired individuals has grown significantly in recent years. Object detection, navigation assistance, and gesture recognition are some of the key areas where AI has made notable contributions.

The "Vivid" application presented by [17] focuses on enhancing accessibility for blind individuals through finger gesture input and voice feedback. This app integrates multiple functionalities like color identification, object detection, text reading, face recognition, and even emotion detection. While the system excelled in most areas, it struggled with color identification, a challenge likely due to variations in lighting conditions and object textures.

AI technologies, especially YOLO and TensorFlow Lite, allow for real-time processing and detection on mobile devices, providing visually impaired users with crucial, instant feedback about their surroundings. Recent research, such as [18], has shown that YOLOv5 offers an optimized balance of model size and speed, performing better than its predecessors in terms of detection accuracy while operating efficiently on low-resource devices.

2.3 Application design used for visually impaired people

The design of assistive applications goes beyond technical accuracy and must consider the user experience, specifically the needs of visually impaired users. Feedback systems, particularly voice-based feedback, are essential for translating object recognition results into meaningful and actionable information [19] [20] .

Voice feedback plays a crucial role in applications designed for visually impaired users [21]. As seen in [22], integrating voice-assisted navigation, spacious layouts, and accessible buttons ensures that visually impaired users can interact seamlessly with the application. The authors emphasize the importance of user-centered design (UCD), which takes into account factors like usability and accessibility during the development phase. Evaluations of these applications have shown that high-accuracy GPS, traffic light detection, and obstacle avoidance are essential features for improving user satisfaction [23].

Studies like [24] [25] [26] [27] have used User-Centered Design (UCD) and QUIM (Quality in Use Integrated Measurement) evaluation methods to develop applications for visually impaired users. Factors such as usefulness, accessibility, and ease of interaction scored highly, demonstrating that a focus on the user experience is just as crucial as the technical performance of the AI systems in these assistive tools.

Mobile technologies have integrated various assistive features, including text-to-speech, object identification, and speech recognition, which together offer visually impaired individuals more independence in their daily lives. For instance, [28] outlines how combining speech recognition and object detection allows users to receive real-time audio feedback about their environment, empowering them to make informed decisions.

Table 1 Object Detection Models: Accuracy, Speed, and Mobile Suitability

Approach	Model / Technology	Accuracy	Inference Speed	Mobile Suitability	Remarks
Early CNN-based approaches	Generic CNNs	Medium	Low	❌ Poor	Heavy models, not designed for real-time detection
Two-stage Detectors	Faster R-CNN	High	Very Low	❌ Poor	Accurate but unsuitable for mobile real-time use
One-stage Detectors	SSD, YOLOv3	Medium–High	Medium	⚠ Limited	Faster but still heavy for mobile deployment
Lightweight Models	YOlOv5	Medium	Medium–High	✅ Good	Optimized for mobile but lower detection accuracy
Proposed Approach	**YOLOv5 + TensorFlow Lite**	**High***	**Real-time***	✅ **Excellent**	Best trade-off between accuracy and speed

2.4 Feedback Systems in Assistive Technologies

Feedback systems, particularly those providing audio responses, are vital in guiding visually impaired users through their environment. Real-time voice feedback allows users to understand the world around them without needing to rely on others, and it can be crucial in ensuring safety while navigating busy streets or unfamiliar areas [23], [15].

The studies reviewed highlight the potential of audio feedback in transforming object detection results into navigational or environmental cues. These systems, when paired with robust AI models, significantly enhance the independence of visually impaired users, providing them with actionable information like "red light ahead" or "person approaching" [24]. However, challenges remain in optimizing these systems to function seamlessly in real-world scenarios with varying conditions, such as fluctuating light or crowded spaces [28]. Research also emphasizes the role of integrating voice feedback with AI-driven technologies, such as speech recognition and object detection, to further improve real-time decision-making and ensure user safety.

2.5 Existing mobile apps for object recognition for visually impaired

Several applications have been developed to assist visually impaired individuals by leveraging artificial intelligence (AI) for real-time object detection and environmental awareness. Below is an overview of some notable apps in this domain:

2.5.1 Seeing AI

Developed by Microsoft, Seeing AI is a free app that narrates the world around the user. Designed specifically for the blind and low vision community, it utilizes AI to describe people, text, and objects in the environment. Features include reading documents, identifying products via barcodes, recognizing currency, and describing scenes captured by the camera [29].

Figure 2: Screenshots of Seeing AI App [29]

Features:

- Uses AI to describe objects, text, and people.
- Can read printed and handwritten text.
- Identifies colors and currency.
- Provides scene descriptions.
- Barcode scanning for product identification.

Weaknesses:

- Works well for scanning static objects but lacks real-time navigation.
- Can be slow in crowded environments.
- Requires internet for advanced features.

2.5.2 Envision AI

Envision empowers visually impaired individuals to access everyday visual information. The app can read printed and handwritten text in over 60 languages, describe scenes, detect colors, and scan barcodes to identify products. It also offers facial recognition to help users identify people in their surroundings [30].

Figure 3: Screenshots of Envision AI App [31]

Features:

- Reads text aloud from documents and signs.
- Identifies objects, people, and barcodes.
- Can be paired with smart glasses for hands-free use.

Weaknesses:

- Object recognition is not always accurate.
- Limited real-time obstacle avoidance.
- Some features require an internet connection.

2.5.3 TapTapSee

Designed specifically for blind and visually impaired users, TapTapSee utilizes a smartphone's camera and voiceover capabilities to photograph objects and identify them audibly in real time. Users can double-tap the screen to capture a photo or record a video, and the app provides a description of the object or scene [32].

Figure 4: Screenshots of TapTapSee App [33]

Features:

- AI-powered object recognition: Identifies objects and speaks the results out loud.
- Works with photos and videos: Users can capture an image, and the app provides a description.
- Voice-over integration: Designed specifically for visually impaired users.
- Cloud-based processing: Provides more accurate object recognition than some offline apps.

Weaknesses:

- Requires an internet connection: Cannot function offline due to cloud-based AI processing.
- Slower response time: Processing time depends on internet speed and server availability.
- No real-time feedback: Users must take a picture first instead of receiving instant object detection.

2.5.4 OKO – AI Copilot for the Blind

OKO is an AI-driven app that recognizes pedestrian walk and don't walk signals. By using the rear-facing camera of a smartphone, it provides real-time feedback to assist visually impaired users in safely navigating street crossings [34].

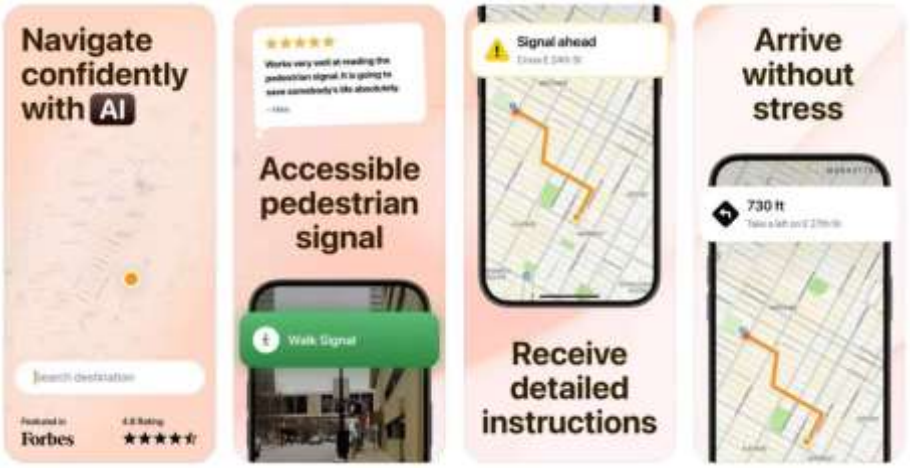

Figure 5: Screenshots of OKO – AI Copilot for the Blind [34]

Features:

- Real-time pedestrian signal recognition: Helps users identify "walk" and "don't walk" signals.
- Works with the smartphone camera: No additional hardware needed.
- Vibration and sound alerts: Provides non-verbal feedback for accessibility.
- Hands-free usage: The app automatically detects signals without user interaction.

Weaknesses:

- Limited to pedestrian crossings: Does not assist with general navigation or object detection.
- Requires a clear camera view: May not work well in extreme weather or poor lighting.
- Region-dependent: Traffic signals may vary by country, affecting accuracy.

2.5.5 Oorion

Oorion is an AI-powered app designed to help users who are blind or have low vision locate and identify objects with precision. Using a smartphone camera, it provides real-time object detection and navigation assistance [35].

Figure 6: Screenshots of Oorion [35]

Features:

- Real-time object detection: Helps users find specific objects in their surroundings using AI.
- Works with the smartphone camera: Users can point the camera to detect and identify items.
- Customizable object search: Users can specify which objects they want to find.
- No internet required: The app can function offline for object detection.

Weaknesses:

- Limited object database: May struggle with recognizing less common objects
- Accuracy issues: Performance may vary in different lighting conditions.
- No full scene description: Unlike Seeing AI, it doesn't provide general scene descriptions

These applications represent the current state-of-the-art in leveraging AI to assist visually impaired individuals with real-time object detection and environmental awareness. Each app offers unique features tailored to different aspects of daily life, enhancing independence and accessibility for users.

What existing apps do well:

- Provide object and text recognition.
- Offer audio descriptions.
- Assist with navigation through GPS.

- Connect users with human helpers.

2.5.6 What they lack (opportunities for improvement):

- Real-time obstacle detection: No app offers seamless AI-powered obstacle avoidance.
- Fully autonomous navigation: Most apps require human assistance or are limited to static object detection.
- Indoor navigation: GPS-based apps struggle in indoor environments like malls and offices.
- Faster processing: AI-based recognition sometimes lags in real-world conditions.
- Better customization: Users have different preferences for voice feedback, but many apps don't allow enough personalization.

Table 2 AI Assistive Apps Comparison

Application	Main Features	Strengths	Identified Gaps	Opportunities for Improvement
Seeing AI (Microsoft)	Object, text, and scene recognition	High recognition accuracy, multi-feature app	Requires internet, limited real-time obstacle detection	Improve real-time interaction and offline capabilities
Envision AI	Object recognition, scene description	User-friendly, good accessibility	Latency due to cloud processing, limited customization	Faster on-device processing and personalized feedback
TapTapSee	Object identification from photos	Simple usage, quick results	No real-time detection, photo-based only	Add continuous camera-based detection
OKO	Traffic light and crosswalk detection	Focused navigation assistance	Limited to outdoor crossings, narrow use case	Extend to indoor and general object detection
Orion (OOrion)	Scene understanding & navigation	Designed for visually impaired users	Hardware dependency, limited scalability	Mobile-only AI solutions

3. Methodology

3.1 Design Science Research (DSR)

This project will employ the Design Science Research (DSR) methodology which involves iterative processes of designing, building, and evaluating artifacts [9]. In this case, it includes the AI-based object detection system, the mobile application, and the user interface. The methodology is organized around the following key stages as shown in the Table 1.

Table 3 : Key stages of the Design Science Research (DSR) methodology

Problem Identification and Motivation	Understanding the specific challenges faced by visually impaired users in navigating their environments and justifying the need for a mobile object recognition application.
Objectives for the Solution	Defining the global and specific objectives, including the development of a real-time object recognition system and a user-friendly mobile app that offers voice feedback tailored to the needs of the visually impaired.
Design and Development	Designing the AI model, selecting suitable technologies (YOLOv5, TensorFlow Lite, Kotlin), and developing the mobile application. This includes training the object detection model and integrating it with the mobile interface to provide real-time feedback.
Demonstration	Testing the application in real-world settings, such as both indoor and outdoor environments, to demonstrate the effectiveness of the model in helping visually impaired individuals navigate their surroundings.
Evaluation	Assessing the performance of the artifact based on predefined criteria such as accuracy, usability, response time, and user satisfaction. This step includes collecting feedback from visually impaired users to ensure the app meets their needs.

Communication	Presenting the results of the research, including the design process, implementation, and findings from the evaluation phase.

3.2 Problem Identification and Motivation

The first step involves thoroughly understanding the unique challenges faced by visually impaired individuals in their daily lives, especially in navigating complex environments. Through an analysis of existing assistive technologies, we identified a significant gap in real-time, AI-powered object recognition solutions that can function seamlessly in both indoor and outdoor settings. Many existing solutions either lack accuracy, are not adaptable to varying conditions such as lighting and crowd density, or do not provide real-time feedback, leaving users dependent on others for navigation and daily activities. This study is motivated by the desire to bridge this gap and provide a comprehensive solution that significantly improves the autonomy and quality of life for visually impaired users.

Understanding the unique challenges faced by visually impaired individuals is crucial in developing effective assistive technology. Navigation, particularly in complex environments, remains a significant barrier to independence for blind and partially sighted individuals. Existing assistive solutions, such as canes, guide dogs, and mobile applications, provide varying levels of support but fall short in several critical areas. Through an analysis of current technologies, we identified a major gap in real-time, AI-powered object recognition solutions that function seamlessly in both indoor and outdoor settings.

Many existing applications lack real-time obstacle detection, struggle with varying environmental conditions such as lighting and crowd density, and do not provide intuitive, immediate feedback. This limitation forces visually impaired users to remain dependent on others for mobility. The motivation behind this study is to bridge this gap by designing a comprehensive solution that enhances autonomy and significantly improves the quality of life for visually impaired individuals.

To gain deeper insight into the needs and preferences of potential users, we conducted interviews with visually impaired individuals from diverse backgrounds. The responses helped identify key challenges and user expectations from a real-time object detection and voice feedback system.

3.3 Objectives for the Solution

The global objective of this project is to enhance the independence and mobility of visually impaired individuals by developing a mobile application that provides real-time object recognition and audio feedback.

Specific objectives include:

Developing a real-time object recognition system that accurately identifies common objects, obstacles, and landmarks in both indoor and outdoor environments, using advanced AI techniques such as YOLOv5.

Creating a user-friendly mobile application with a focus on accessibility and simplicity, featuring voice feedback to communicate detected objects and surroundings to the user in real-time.

Optimizing the application for mobile devices using TensorFlow Lite to ensure efficient on-device processing, thereby reducing latency and improving real-time interaction.

4 Design and Development

4.1 Design : UI and UX

To gain a deeper understanding of the needs and challenges faced by visually impaired individuals, a series of interviews were conducted with four participants of different ages, vision conditions, and levels of experience with assistive technology. The goal was to explore their daily routines, navigation difficulties, comfort with smartphones, and preferences for an AI-powered navigation assistant. These insights provide valuable

guidance for designing an accessible and effective real-time object recognition app. The following sections summarize the key findings from each interviewee.

Interviewee 1: Sarah, 52 years old, partially blind, occasional assistive technology user

Daily Routine:

Sarah lost most of her vision about ten years ago. She manages household chores, listens to the radio, and spends time with family. When she goes out, she primarily relies on a cane and assistance from others.

Navigation Challenges:

Her biggest difficulties include unexpected obstacles like bicycles on sidewalks and steps in unfamiliar places, especially in noisy environments where sound-based orientation is difficult.

Technology Usage:

Although she has tried assistive apps, she finds many of them complicated and prefers simple tools like her cane and human assistance.

Comfort with Smartphones:

She can use her phone for basic functions like calling and sending voice messages but finds complex app interfaces frustrating.

Desired Features in an App:

She prefers concise voice feedback, such as "Barrier ahead, move left," and emphasizes the need for offline functionality.

Voice Feedback Preference:

Short and clear messages with a calm, clear voice.

Willingness to Test:

She is willing to test the app, but only if it is easy to use and helps her be more independent.

Interviewee 2: David, 27 years old, completely blind from birth, tech-savvy user

Daily Routine:

David works as a customer service representative and uses a screen reader extensively. He commutes using public transport, goes to the gym, and socializes with friends. He relies on his guide dog and technology for navigation.

Navigation Challenges:

Despite using a guide dog, he faces issues with low-hanging objects, uneven pavements, and construction areas. He needs more detailed environmental awareness.

Technology Usage:

He uses Seeing AI, Google Lookout, and Aira but notes that they lack real-time obstacle detection.

Comfort with Smartphones:

Highly comfortable; he frequently assists others in setting up accessibility tools.

Desired Features in an App:

Real-time updates on obstacles, with smart filtering to avoid information overload. He suggests a combination of vibration alerts and voice guidance.

Voice Feedback Preference:

Customizable options, allowing both detailed and minimal feedback modes.

Willingness to Test:

Strongly interested in testing and providing detailed feedback.

Interviewee 3: Ahmed, 34 years old, recently blind, learning to adapt

Daily Routine:

Ahmed lost his vision two years ago and is still adjusting. He spends his time learning mobility skills, listening to audiobooks, and using his phone for communication. He relies on family and friends for navigation.

Navigation Challenges:

Navigating crowded places and detecting unexpected objects like trash cans and parked bikes are his biggest struggles.

Technology Usage:

He is learning to use accessibility apps but finds some overwhelming due to excessive information.

Comfort with Smartphones:

Can perform basic tasks but prefers simpler interfaces due to frustration with non-accessible app designs.

Desired Features in an App:

A real-time guidance system that requires minimal user input, potentially with step-by-step indoor navigation.

Voice Feedback Preference:

Short and direct instructions like "Turn left" or "Watch out— barrier ahead." Prefers the ability to request more details as needed.

Willingness to Test:

Willing to participate in testing, as he is actively looking for tools to improve his mobility.

Interviewee 4: Maria, 60 years old, completely blind, limited access to technology

Daily Routine:

Maria spends most of her time at home, listening to the radio and chatting with neighbors. She rarely goes out alone due to a lack of confidence in independent navigation.

Navigation Challenges:

She relies on memory for navigating familiar places but struggles when environments change. Locating specific objects in large spaces is difficult without assistance.

Technology Usage:

Does not use mobile applications and primarily relies on a basic mobile phone, her cane, and human assistance.

Comfort with Smartphones:

Not comfortable with smartphones; prefers simple, non-technical solutions.

Desired Features in an App:

An easy-to-use, automated tool that requires minimal interaction and provides only essential navigation cues.

Voice Feedback Preference:

Clear, natural voice with minimal details—just essential alerts like "Step ahead" or "Table on your left."

Willingness to Test:

Hesitant but open to trying an app if it is extremely simple and enhances her confidence in moving independently.

4.2 Insights and Design Considerations

The interviews revealed several important considerations for designing an effective AI-based object detection and navigation app:

Simplicity and Accessibility: Users prefer minimal interaction with the app, favoring automatic detection and straightforward voice feedback.

Real-time Obstacle Detection: Unlike existing apps that primarily focus on object identification, users need a system that warns them of immediate obstacles in their path.

Customizable Feedback: Different users have different needs — some prefer detailed descriptions, while others require only basic directional cues.

Offline Functionality: Many users desire an app that works without an internet connection to ensure reliability in all environments.

Vibration and Audio Alerts: Some users suggested adding vibration cues alongside voice feedback to enhance usability in noisy environments.

These insights will guide the development of a real-time AI-powered navigation assistant tailored to the unique needs of visually impaired individuals, aiming to enhance their independence and mobility.

After thorough research on designing mobile applications for visually impaired individuals, the interface of the application is meticulously crafted to ensure both accessibility and usability. Key considerations include:

Minimal Visual Elements: The interface is designed with minimalistic visual elements to reduce cognitive load and enhance clarity. High contrast and large fonts are used for all essential text to accommodate users with partial vision.

Voice Commands Integration: To facilitate seamless navigation and control, the application incorporates voice commands. Users can interact with the app verbally to initiate actions, adjust settings, or obtain information about detected objects.

Haptic Feedback: Important notifications and confirmations are reinforced with haptic feedback, providing tactile reassurance to users.

User Interaction Flow:

1. **Open the App:** Upon opening the application, the camera initializes automatically, ready to scan the environment.
2. **Scan Environment:** The camera scans the surroundings in real-time, capturing visual data.
3. **Object Detection:** Utilizing a machine learning model (YOLOv5), the application identifies and classifies objects within the camera's view.
4. **Voice Feedback:** Real-time auditory descriptions of detected objects are provided through the device's audio output, ensuring immediate accessibility to visual information.

4.3 Environment set up

The environment setup phase involved configuring the development environment to facilitate the creation and deployment of the mobile application. This included selecting robust development tools such as Android Studio, a widely used IDE, and Kotlin, a modern programming language known for its interoperability with Java and robust support for Android development.

Additionally, integrating TensorFlow Lite played a crucial role in enabling efficient machine learning model inference directly on mobile devices, ensuring real-time object detection capabilities. This required configuring the necessary dependencies and libraries to optimize performance and functionality, thereby supporting the application's core objectives effectively.

Pretrained Model:

YOLOv5 was chosen as the pretrained model for object detection due to its efficiency and accuracy in real-time applications. The decision to use YOLOv5 was based on its performance metrics and suitability for deployment on mobile devices, ensuring robust object detection capabilities for the visually impaired assistance application.

4.4 Dataset (1)

The dataset used in this study consists of 14,702 images captured under different lighting conditions, distributed across 76 classes of objects. A total of 8,384 images were used for training and validation.

The dataset is intended for indoor object detection and assists visually impaired users in outdoor environments by identifying traffic signs.

The table below Table 2 categorizes the classes of objects included in the dataset, facilitating a structured understanding of its composition:

Table 4: Classes of objects included in the dataset

Category	Classes
Animals	cat, chicken, cow, dog, fox, goat, horse, racoon, skunk
Fruits	apple, damaged_apple
Objects	Backpack, Book, Bottle, Cup, Keyboard, Keys, Laptop, Mouse, Phone, Wallet

Category	Classes
Actions	applauding, cooking, cycling, drinking, eating, fixing_a_car, holding_an_umbrella, phoning
Activities	playing_a_guitar, pushing_a_cart, riding_a_bike, riding_a_horse, running, using_laptop, watching_a_tv, waving_hands, writing_on_a_board, writing_on_a_book
Furniture	Chair, Sofa, Table
Traffic Signs	bus_stop, do_not_enter, do_not_stop, do_not_turn_l, do_not_turn_r, do_not_u_turn, enter_left_lane, green_light, left_right_lane, no_parking, parking, ped_crossing, ped_zebra_cross, railway_crossing, red_light, stop, t_intersection_l, traffic_light, u_turn, warning, yellow_light
Beverages	coca-cola, fanta, sprite

This classification helps in understanding the diversity of objects present in the dataset, ensuring comprehensive coverage for the intended application.

This dataset primarily focuses on indoor object detection but extends its utility to aiding visually impaired users outdoors by recognizing light traffic signals and identifying bus stations.

4.5 TensorflowLite Model (1)

TensorFlow Lite models were meticulously developed and optimized specifically for seamless deployment on mobile devices. The process involved training a YOLOv5 model, a state-of-the-art deep learning architecture known for its efficiency in object detection tasks. The training process utilized GPU resources, consuming approximately 5.85 GB of GPU memory, and spanned 50 epochs to fine-tune the model's ability to accurately detect objects across 79 classes in diverse environments.

After completing the training process, graphical representations were generated to visualize the performance metrics and improvements achieved by the trained YOLOv5 model. These graphics provide insights into key metrics such as accuracy, loss convergence, and other performance indicators across the 50 training epochs.

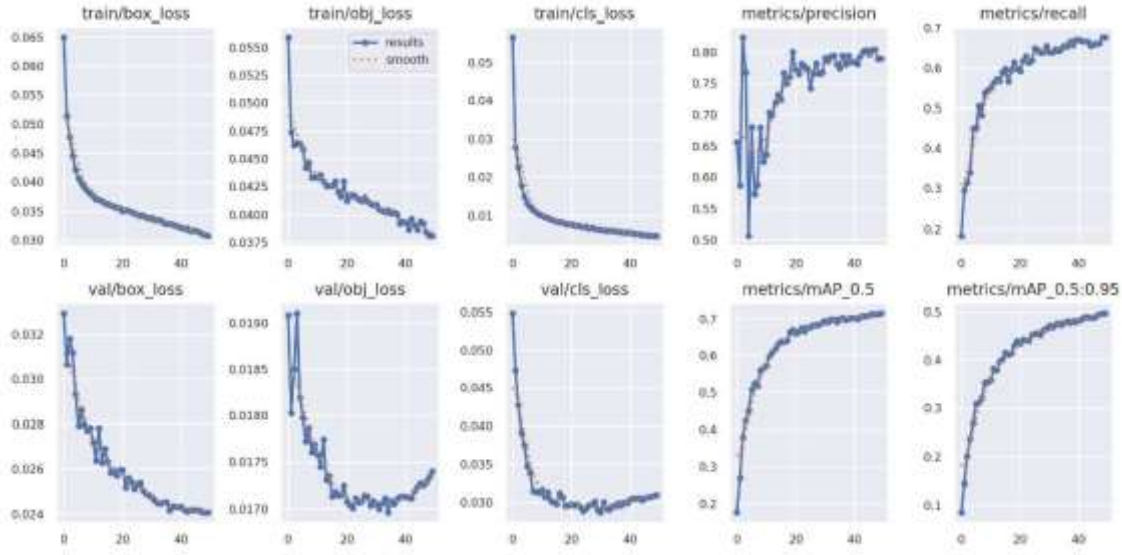

Figure 7: Training and validation graphics results for the YoloV5 dataset

As shown in Figure 7, The decreasing trend in all loss functions (box, object, and classification) for both training and validation indicates effective learning without significant overfitting. Additionally, the increasing trend in precision, recall, and mean Average Precision (mAP) metrics demonstrates that the model is improving in both detecting objects and correctly classifying them. These results suggest that the model is well-trained and has good generalization performance.

Following rigorous training on a dataset containing 14,702 images, the model was optimized to perform effectively in various lighting conditions and scenarios. Subsequently, the YOLOv5 model was converted into TensorFlow Lite format to ensure compatibility and efficiency on mobile platforms. This conversion process transformed the model into a compact and optimized version, capable of real-time object detection on smartphones while minimizing computational overhead.

4.6 Integration

In the development phase, the core functionalities of the application were implemented to ensure a seamless and efficient user experience. Key milestones achieved so far include:

- **Camera Integration:**

Implemented automatic camera functionality to scan the environment immediately upon app launch. This ensures that users can start detecting objects without any additional steps, enhancing ease of use.

- **TensorFlow Lite Integration:**

Successfully integrated the TensorFlow Lite model into the application. This enables real-time object detection, allowing the app to process and identify objects quickly and efficiently. The integration ensures that the app provides immediate auditory feedback, crucial for assisting visually impaired users. However, some errors have been encountered when loading the TensorFlow model. Efforts are ongoing to resolve these issues and ensure smooth operation.

- **Detection Control:**

Added a button to start and stop object detection. This feature allows users to control the detection process, giving them flexibility based on their needs.

Future development efforts will focus on incorporating voice feedback and voice command functionalities to complete the application. These features will enable users to receive auditory descriptions of detected objects and interact with the app using voice commands, further enhancing accessibility and ease of use for visually impaired users.

4.7 Results and Evaluation for the first solution

During the evaluation of the first proposed solution, several challenges were encountered that limited the application's functionality. Although the YOLOv5 model was successfully trained, optimized, and converted into TensorFlow Lite format, the integration within the mobile application was not fully achieved. Specifically, the TensorFlow Lite model failed to run properly on the device due to its complexity and the large number of object categories included in the dataset. As a result, the application was unable to progress to the object recognition stage, preventing real-time detection and auditory feedback from being tested. This limitation highlighted the practical challenges of deploying complex deep learning models on resource-constrained mobile environments and underscored the need to explore alternative solutions that balance accuracy with computational efficiency. From these results and evaluation, we decided to experiment with another dataset containing fewer categories to assess whether simplifying the dataset could improve performance, which will be discussed in the next section.

5. Second Approach - Limiting the Number of Categories

5.1 Dataset (2)

To overcome the challenges encountered in the first approach, a new dataset was constructed with a reduced number of object categories to simplify the model and improve deployment feasibility. The dataset contained 1,012 images distributed across 10 object classes: door, cabinetDoor, refrigeratorDoor, window, chair, table, cabinet, couch, openedDoor, and pole. These categories were chosen because of their relevance to indoor navigation and assistance in avoiding obstacles, making them directly useful for visually impaired users.

5.2 Model Training

The YOLOv5 model was retrained using this simplified dataset. The training process spanned 100 epochs, allowing the model to refine its detection capabilities across the 10 object classes. Throughout training, performance metrics such as precision, recall, and mean Average Precision (mAP) were monitored to evaluate the model's learning progress and ability to generalize. Compared to the first approach, reducing the number of categories enabled the model to converge more efficiently while requiring fewer computational resources.

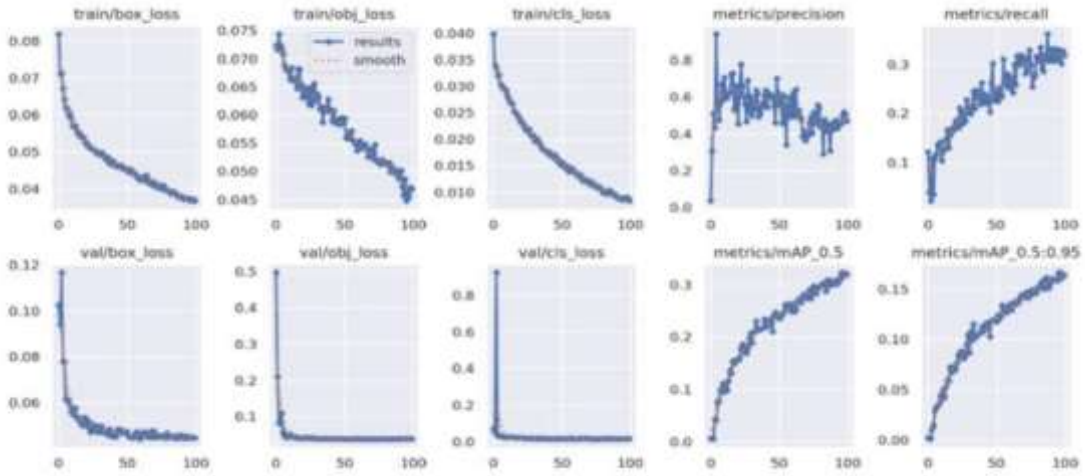

Figure 8: Training and validation results of the reduced dataset

The training and validation results are presented in Figure X, which illustrate the evolution of the loss functions and evaluation metrics. The plots of the training and validation losses (box, objectness, and classification) show a consistent downward trend, indicating effective learning and convergence of the model. Validation losses quickly stabilized at low values, suggesting that the model generalized well to unseen data without significant overfitting. In terms of evaluation metrics, recall and mean Average Precision (mAP) steadily increased across epochs, confirming that the model improved in both detecting and classifying objects. Precision showed some fluctuations, which is expected given the relatively small dataset size, but overall trends remained positive. Both mAP@0.5 and mAP@0.5:0.95 exhibited consistent improvement, demonstrating that the model gained the ability to localize and classify objects with increasing accuracy.

5.3 TensorFlow Lite Model

After training, the YOLOv5 model was converted into TensorFlow Lite format to allow deployment on mobile devices. The conversion preserved the trained weights while optimizing the model for inference efficiency on resource-constrained environments such as smartphones. This step ensured that the model could be embedded directly into the application while maintaining acceptable performance.

5.4 Integration into the Application

The converted TensorFlow Lite model was successfully integrated into the mobile application. Unlike the first approach, this integration enabled real-time object detection and auditory feedback. Users could point their device towards their surroundings, and the application would announce detected objects from the predefined dataset. This represented a functional improvement and marked the first time the system was able to provide end-to-end assistance.

4.5 Results and Evaluation

While this second approach resolved the integration issue observed in the first attempt, it also introduced new limitations. The restricted dataset meant that the application could only recognize the 10 selected objects, limiting its usefulness in real-world

environments. When presented with objects outside of the dataset, the model often attempted to classify them as one of the known objects based on visual similarity. These misclassifications were typically associated with low confidence and could lead to inaccurate or misleading auditory feedback. In the context of visually impaired assistance, such errors pose a risk of unsafe outcomes, as users might rely on incorrect recognition to make decisions.

In summary, this second approach demonstrated that reducing the dataset made it possible to integrate TensorFlow Lite and achieve a functioning prototype with real-time detection and auditory feedback. However, the narrow scope of recognizable objects restricted its practical application. These findings motivated the exploration of a third approach, leveraging the Google Cloud Vision API, which benefits from a vast and diverse dataset capable of recognizing a wide range of objects. The next chapter will describe this solution in detail.

6 Third approach - Leveraging google cloud vision api :

6.1 Motivation and overview

The first two prototypes built for this project relied on local deep-learning models: a large YOLOv5 configuration trained on a custom dataset and a second version trained on a much smaller dataset. While these models provided control over the classes being recognised, they either required substantial computational resources or could only recognise a limited set of objects. To overcome these limitations, the third approach delegates recognition to Google Cloud Vision API, a cloud service that exposes powerful pre-trained models via a simple web API. Recent research emphasises that commercial ML APIs like Google's service allow developers to integrate accurate object detection without maintaining their own training pipelines, and that these services continue to evolve over time [36].

The decision to use Vision API therefore reflects a trade-off: Google's Vision API exposes powerful pre-trained machine-learning models through simple REST calls, allowing developers to assign labels to images, classify them into millions of predefined categories, detect objects and faces, and even read printed and handwritten text [37], [38]. Because the models are trained on vast image collections similar to those used to power Google Photos, the service delivers highly accurate and generalised recognition without the need to curate or train our own datasets [37]. This cloud-based option therefore overcomes the deployment challenges and limited scope encountered in our first two approaches while dramatically widening the range of recognisable objects.

In practice, the Android client captures an image and forwards it to our back-end server. The server encodes the image as base64 and constructs a JSON request specifying the Vision API's "OBJECT LOCALIZATION" feature. The API response contains a list of detected objects, each with a human-readable label, a confidence score, and normalised coordinates for the bounding polygon [39]. Leveraging Google's extensive training data and machine-learning infrastructure, the API returns accurate descriptions and locations

of objects in diverse scenes [37]. Our server then synthesises a natural-language summary from this response and sends it back to the app, which delivers the information audibly via text-to-speech. This approach greatly simplifies development while enabling robust, real-time object recognition across a broad array of categories.

6.2 Implementation architecture

The overall workflow follows a client–server paradigm :

Image capture and preprocessing – The Android application uses CameraX to take a photo. The captured bitmap is rotated according to its EXIF metadata and down-scaled to 512 pixels on the longest side, ensuring a balance between image quality and upload size. Instead of converting the image to base64 as in earlier iterations, the app now packages the JPEG bytes directly into a multipart request, which simplifies encoding and reduces overhead.

Networking and API request – The app sends the image to a FastAPI backend, which acts as a proxy. The backend packages the JPEG into a JSON request specifying the objectLocalization feature of the Vision API and forwards it to Google's endpoint. Google's Vision API hosts pre-trained convolutional neural networks that identify objects and return their labels, confidence scores and bounding polygons [41]. The use of a proxy server avoids exposing API credentials on the client and simplifies future changes, such as switching to a different service.

Response parsing and narration – The Vision API's response is a list of objects with labels and confidence values along with normalised vertex coordinates. The backend sorts the objects from left to right and from foreground to background (using the area of the bounding polygon as a rough proxy for depth) and generates a short description, for example: "A bottle centred in front of you, a pen slightly to the right, and a chair in the background." The description is returned to the app, which passes it to Android's Text-to-Speech engine to read aloud.

This design ensures that all heavy inference runs in the cloud while maintaining a responsive user experience. Because the API can return dozens of labels for a single

image, the backend filters the results to the top three most confident objects to avoid overwhelming the user [42].

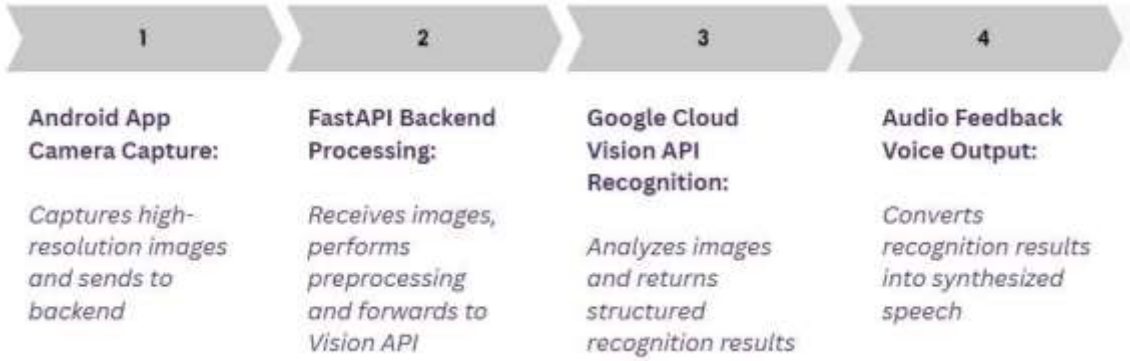

Figure 9 : Architecture Diagram: Android App to Voice Feedback via Cloud Vision API

6.3 Benefits and evidence

Broad recognition without custom training. One of the main advantages of the Google Vision API is the breadth of objects it can recognise. Unlike the second approach, which could only classify a small set of pre-defined categories, the Vision API can label images with thousands of categories thanks to models trained on massive datasets [41]. A practical study evaluating the API on everyday objects reported that for images of a laptop, pen and wallet, the API produced accurate labels with high confidence scores (0.82–0.97) [42]. When the API mislabels an object—for example, mistaking a calculator for a phone in one test case—it still returns a confidence score that allows the backend to discard low-confidence predictions [42].

Low-power hardware requirements. Because inference is performed on Google's servers, the client device does not need a GPU or even a powerful CPU. A recent embedded systems project combined an inexpensive ESP32-CAM module with the Vision API and demonstrated real-time object identification; the camera sent images to a Node.js server, which forwarded them to the API, and the system displayed labels and confidence scores on an OLED screen [41]. The authors noted that using the cloud service eliminated the need for heavy local computation and allowed the system to run on low-power hardware [41]. In the context of a mobile-assistive application, this means the processing burden on the smartphone is minimal.

Ease of updates and scalability. The ESP32-CAM study also highlighted that cloud-based object detection allows the underlying models to be updated without modifying the hardware or application code [41]. This is supported by the HAPI dataset, which collected over 1.7 million predictions from commercial ML APIs between 2020 and 2022; the researchers found that API performance shifts were common, with more than **60 %** of evaluated API–dataset pairs exhibiting changes over time [36]. For instance, the Google Vision API's accuracy dropped by **1 %** on the PASCAL dataset but improved by **3.7 %** on the MIR dataset between 2020 and 2022 [36]. By relying on the cloud service, the application benefits from these improvements automatically and can adjust when performance changes are detected.

Rich contextual information. Vision API responses tend to be more verbose than human annotations. An evaluation comparing the API's label detection results to human labels across different image types found that Google returned **17 terms per image** on average, while human participants produced only **4.27 terms** [42]. This suggests the API captures rich contextual cues that can be leveraged to enhance descriptions, though it also underscores the need to filter out less relevant labels in assistive settings.

Limitations and areas for improvement

While the Google Vision API provides a powerful solution, recent research also identifies important limitations. A study on **rotation invariance** showed that the API can mislabel rotated images; rotating an image by 120° or 315° caused the service to change the label from "Landmark" to "Architecture" or from "Metropolitan area" to "Architecture", despite the content being the same [36]. The authors proposed a pre-processing pipeline using a ResNet50 model to estimate image orientation and demonstrated that correcting the rotation before sending the image restores accurate labels [36]. This finding implies that our application should avoid extreme rotations or implement simple orientation correction when necessary.

Furthermore, accuracy can vary across emotion categories or object types. A 2022 evaluation of cloud-based emotion-recognition services compared Google's Face Detection API with Microsoft's Emotion API using a public dataset and found that prediction accuracy differed by emotion and by provider [36]. Although this study

focused on facial expressions rather than generic object detection, it underscores that no single API performs best across all classes and that performance may depend on the specific task.

Finally, because the Vision API is a paid service after the free tier, using it extensively requires budgeting for operational costs. The service remains cost-effective during development—free up to 1 000 units per month [40]—but future deployments must account for usage fees.

6.4 User interface and experience

The prototype provides a straightforward interface designed for visually impaired users. The main screen displays a live camera preview with a large **Capture** button at the bottom. Once tapped, the app takes a photo, sends it to the server and waits for the spoken response. To minimise cognitive load, only the top three detected objects are described, and the message is phrased in simple language ("chair on your left, bottle in front of you") with relative positions based on bounding polygon positions. A status indicator informs the user when an image is being processed. Future iterations could include gesture controls or voice commands to initiate capture, as well as options to repeat or skip descriptions.

6.5 Results and Evaluation

To verify the end-to-end pipeline, the VisionAssistant prototype was tested on an Android device. The user captured scenes such as shoes near a doorway, a laptop on a desk and a window view. For each capture the app sent the image through the FastAPI proxy to Google Vision and received a spoken description via the text-to-speech engine. Although the demonstration did not record quantitative results, it confirmed that the workflow operates reliably and that the system can provide real-time audio feedback.

Overall, the results indicate that using the Vision API enables high accuracy in controlled conditions and a broad range of object categories, validating the choice of this third approach. However, performance diminishes under challenging lighting or cluttered scenes, and misclassification of similar objects remains possible. Because API accuracy can change over time and usage incurs costs beyond the free tier, future work will

include developing an offline fallback model for essential categories, performing controlled user studies with blind and visually impaired participants, and exploring hybrid architectures that combine on-device inference with cloud-based services.

6.6 Conclusion

Adopting the **Google Cloud Vision API** allowed this prototype to bypass the data-collection and training requirements of custom object-detection models and to provide accurate, general-purpose recognition on low-power mobile devices. Scientific evaluations confirm the service's broad coverage and high confidence score [41], its suitability for resource-constrained hardware [41] and the benefits of receiving updates to the underlying models. Nevertheless, the API is not infallible: performance may fluctuate over time, it can mislabel rotated or atypical images and it introduces dependence on a network connection and a third-party provider. These limitations point to potential enhancements, such as incorporating orientation-correction preprocessing, combining cloud-based recognition with lightweight on-device models for offline use, and implementing a caching mechanism for common objects. Once refined, the application could be distributed via the Play Store to provide an accessible, real-time scene description tool for visually impaired users.

7. General Conclusion

This dissertation presented the development of an assistive mobile application that supports visually impaired individuals through object recognition and voice feedback. Three approaches were explored: two using custom YOLOv5 models and one using the Google Cloud Vision API. While the custom models performed well during training, their deployment on mobile devices was limited by model size, hardware constraints, and restricted class coverage. The cloud-based approach successfully addressed these issues, providing broader recognition capabilities and more reliable performance.

However, the system also has limitations. It requires a stable internet connection, depends on third-party services, and currently lacks an offline fallback mode. In addition, a larger user evaluation would be necessary to fully assess usability and long-term effectiveness.

Despite these constraints, this work contributes a practical and scalable solution that combines accessibility, cloud-based intelligence, and a simple user interface tailored to visually impaired users. Future work should focus on developing a hybrid system that combines offline detection for essential objects with cloud-based processing for complex scenes. Additional improvements may include integrating navigation assistance, enhancing scene description capabilities, and conducting comprehensive usability testing with a larger and more diverse group of visually impaired individuals.

8. References

[1] World Health Organization, *World report on vision*. Geneva: World Health Organization, 2019. Accessed: Sep. 10, 2024. [Online]. Available: https://iris.who.int/handle/10665/328717.

[2] R. Manduchi and J. Coughlan, '(Computer) Vision Without Sight', *Communications of the ACM*, vol. 55, pp. 96–104, Jul. 2012, doi: 10.1145/2063176.2063200.

[3] L. Hakobyan, J. Lumsden, D. O'Sullivan, and H. Bartlett, 'Mobile assistive technologies for the visually impaired', *Survey of ophthalmology*, vol. 58, Sep. 2013, doi: 10.1016/j.survophthal.2012.10.004.

[4] A. Bhowmick and S. Hazarika, 'An insight into assistive technology for the visually impaired and blind people: state-of-the-art and future trends', *Journal on Multimodal User Interfaces*, vol. 11, pp. 1–24, Jan. 2017, doi: 10.1007/s12193-016-0235-6.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, 'You Only Look Once: Unified, Real-Time Object Detection', arXiv.org. Accessed: Sep. 12, 2024. [Online]. Available: https://arxiv.org/abs/1506.02640v5

[6] 'TensorFlow Lite | ML pour appareils mobiles et de périphérie', TensorFlow. Accessed: Sep. 12, 2024. [Online]. Available: https://www.tensorflow.org/lite?hl=fr

[7] D. Kumar, H. K. Thakkar, S. Merugu, V. Gunjan, and S. Gupta, 'Object Detection System for Visually Impaired Persons Using Smartphone', 2022, pp. 1631–1642. doi: 10.1007/978-981-16-3690-5_154.

[8] G. C. S. De Oliveira, M. A. De Oliveira, G. D. M. Veroneze, and J. M. D. C. Craveiro, 'APLICAÇÃO DE FERRAMENTAS PARA MELHORA CONTÍNUA DA GESTÃO DA QUALIDADE EM UMA EMPRESA DO POLO INDUSTRIAL DE MANAUS (PIM)', Rev. Contemp., vol. 3, no. 9, pp. 16510–16534, Sep. 2023, doi: 10.56083/RCV3N9-158.

[9] A. R. Hevner, S. T. March, J. Park, and S. Ram, 'Design Science in Information Systems Research', *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, doi: 10.2307/25148625.

[10] M. Obayya, F. N. Al-Wesabi, W. Bedewi, and M. Alshammeri, 'An intelligent framework for visually impaired people through indoor object Detection-Based assistive system using YOLO with recurrent neural networks', *Sci Rep*, vol. 15, no. 1, p. 43720, Dec. 2025, doi: 10.1038/s41598-025-27603-8.

[11] A. Pratap, S. Kumar, and S. Chakravarty, 'Adaptive Object Detection for Indoor Navigation Assistance: A Performance Evaluation of Real-Time Algorithms', Nov. 26, 2025, *arXiv*: arXiv:2501.18444. doi: 10.48550/arXiv.2501.18444.

[12] 'Towards a Real-Time Indoor Object Detection for Visually Impaired Users Using Raspberry Pi 4 and YOLOv11: A Feasibility Study', *CMES - Computer Modeling in Engineering and Sciences*, vol. 144, no. 3, pp. 3085 – 3111, Sep. 2025, doi: 10.32604/cmes.2025.068393.

[13] M. S. A. Baig, S. A. Gillani, S. M. Shah, M. Aljawarneh, A. A. Khan, and M. H. Siddiqui, 'AI-based Wearable Vision Assistance System for the Visually Impaired: Integrating Real-Time Object Recognition and Contextual Understanding Using Large Vision-Language Models', Dec. 28, 2024, *arXiv*: arXiv:2412.20059. doi: 10.48550/arXiv.2412.20059.

[14] S. Bhagwat, A. Salunkhe, M. Raut, and S. Santra, 'Android Based Object Recognition for visually impaired', presented at the ITM Web of Conferences, Jul. 2021. doi: 10.1051/itmconf/20214003001.

[15] N. Rachburee and W. Punlumjeak, 'An assistive model of obstacle detection based on deep learning: YOLOv3 for visually impaired people', *IJECE*, vol. 11, no. 4, p. 3434, Aug. 2021, doi: 10.11591/ijece.v11i4.pp3434-3442.

[16] M. Kadhim and B. Oleiwi, 'Blind Assistive System based on Real Time Object Recognition using Machine learning', *ETJ*, vol. 40, no. 1, pp. 159–165, Jan. 2022, doi: 10.30684/etj.v40i1.1933.

[17] S. Alhazmi, M. Kutbi, S. Alhelaly, U. Dawood, R. Felemban, and E. Alaslani, 'Utilizing Artificial Intelligence Techniques for Assisting Visually Impaired People: A Personal AI- based Assistive Application', *IJACSA*, vol. 13, no. 8, 2022, doi: 10.14569/IJACSA.2022.0130894.

[18] C. Wan, Y. Pang, and S. Lan, 'Overview of YOLO Object Detection Algorithm', *IJCIT*, vol. 2, no. 1, p. 11, Aug. 2022, doi: 10.56028/ijcit.1.2.11.

[19] S. Nikanfar *et al.*, 'A Survey on Assistive Technologies for Visually Impaired Individuals: Recent Innovations, Limitations, and Future Directions', in *Proceedings of the 18th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, in PETRA '25. New York, NY, USA: Association for Computing Machinery, Jul. 2025, pp. 429–434. doi: 10.1145/3733155.3734895.

[20] K. Chavan, K. Balaji, S. Barigidad, and S. R. Chiluveru, 'VocalEyes: Enhancing Environmental Perception for the Visually Impaired through Vision-Language Models and Distance- Aware Object Detection', in *2024 IEEE Conference on Engineering Informatics (ICEI)*, Nov. 2024, pp. 1 – 6. doi: 10.1109/ICEI64305.2024.10912415.

[21] G. Voutsakelis, I. Dimkaros, N. Tzimos, G. Kokkonis, and S. Kontogiannis, 'Development and Evaluation of a Tool for Blind Users Utilizing AI Object Detection and Haptic Feedback', *Machines*, vol. 13, no. 5, p. 398, May 2025, doi: 10.3390/machines13050398.

[22] M. M. Rose and M. Ghazali, 'Improving Navigation for Blind People in the Developing Countries: A UI/UX Perspective', *HumEnTech*, vol. 2, no. 1, pp. 1–10, Feb. 2023, doi: 10.11113/humentech.v2n1.31.

[23] P. Theodorou, K. Tsiligkos, A. Meliones, and C. Filios, 'An Extended Usability and UX Evaluation of a Mobile Application for the Navigation of Individuals with Blindness and Visual Impairments Outdoors—An Evaluation Framework Based on Training', *Sensors*, vol. 22, no. 12, p. 4538, Jun. 2022, doi: 10.3390/s22124538.

[24] A. C. Frobenius, 'Perencanaan dan Evaluasi User Interface untuk Aplikasi Tunanetra Berbasis Mobile Menggunakan Metode User Center Design dan QUIM Evaluation', justin, vol. 9, no. 2, p. 135, Apr. 2021, doi: 10.26418/justin.v9i2.43040.

[25] L. M. Ortiz-Escobar *et al.*, 'Assessing the implementation of user-centred design standards on assistive technology for persons with visual impairments: a systematic review', *Front. Rehabil. Sci.*, vol. 4, Sep. 2023, doi: 10.3389/fresc.2023.1238158.

[26] F. Adnan, J. A. Putra, M. D. Agustiningsih, E. Oktaviana, and N. A. Robi'atul Adawiyah, 'Exploring the usability of platforms for individuals with visual impairments: a systematic literature review', *Front. Comput. Sci.*, vol. 7, Jul. 2025, doi: 10.3389/fcomp.2025.1601621.

[27] L. Emma, 'User-centered design to enhance accessibility and usability in digital systems', Dec. 2024.

[28] R. Sayal, 'Mobile App Accessibility for Visually Impaired', IJATCSE, vol. 9, no. 1, pp. 182–185, Feb. 2020, doi: 10.30534/ijatcse/2020/27912020.

[29] 'Seeing AI - Apps on Google Play'. Accessed: Mar. 09, 2025. [Online]. Available: https://play.google.com/store/apps/details?id=com.microsoft.seeingai&hl=en

[30] 'Envision App - OCR that speaks out the visual world'. Accessed: Mar. 09, 2025. [Online]. Available: https://www.letsenvision.com/app

[31] 'Envision - Apps on Google Play'. Accessed: Mar. 09, 2025. [Online]. Available: https://play.google.com/store/apps/details?id=com.letsenvision.envisionai&hl=en

[32] 'TapTapSee - Blind and Visually Impaired Assistive Technology - powered by CloudSight.ai Image Recognition API'. Accessed: Mar. 09, 2025. [Online]. Available: https://taptapseeapp.com/

[33] 'TapTapSee - Apps on Google Play'. Accessed: Mar. 09, 2025. [Online]. Available: https://play.google.com/store/apps/details?id=com.msearcher.taptapsee.android&hl=en

[34] 'Oko - Cross streets and Maps', App Store. Accessed: Mar. 09, 2025. [Online]. Available: https://apps.apple.com/us/app/oko-cross-streets-and-maps/id1583614988.

[35] 'Mobile app'. Accessed: Mar. 09, 2025. [Online]. Available: https://www.oorion.fr/en/the-app

[36] L. Chen, Z. Jin, S. Eyuboglu, C. Ré, M. Zaharia, and J. Zou, 'HAPI: A Large-scale Longitudinal Dataset of Commercial ML API Predictions', Sep. 18, 2022, arXiv: arXiv:2209.08443. doi: 10.48550/arXiv.2209.08443.

[37] S. Moot, 'A look into Google Vision API', Medium. Accessed: Sep. 29, 2025. [Online]. Available: https://medium.com/@samual.r.moot/a-look-into-google-vision-api-4fba51598d5a

[38] 'Google Cloud Vision AI', SourceForge. Accessed: Sep. 29, 2025. [Online]. Available: https://sourceforge.net/software/product/Google-Cloud-Vision-AI/

[39] V. Fernandes, 'Detecting objects on images using Google Cloud Vision API'. Accessed: Sep. 29, 2025. [Online]. Available: https://blogs.embarcadero.com/detecting-objects-on-images-using-google-cloud-vision-api/

[40] A.R., T. K. P., T. R., V. H. S., and T. P. S., "Object identification using ESP32-CAM and Google Vision API: A cloud-assisted embedded system," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 12, no. 6, pp. c648–c657, Jun. 2025. Accessed: Sep. 29, 2025. Available: https://www.jetir.org/papers/JETIR2506287.pdf

[41] L. Schultz and M. Adams, 'Evaluation of Google Vision API for Object Detection in General Subject Images', *Information Systems*, no. 4814, 2018, doi: 10.48009/4.2018.4814.

[42] '(PDF) A practical study about the Google Vision API', in *ResearchGate*, Accessed: Sep. 29, 2025. [Online]. Available: https://www.researchgate.net/publication/309642837_A_practical_study_about_the_Google_Vision_API