BATCH NO:12

REAL-TIME OBJECT DETECTION AND AUDIO GUIDANCE FOR VISUALLY IMPAIRED INDIVIDUALS

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Artificial Intelligence and Data Science

By

RAHUL L	(22UEAD0050)	**(VTU 21369)**
RAVIPRASAD G	(22UEAD0052)	**(VTU 23521)**
ANBARASU S	(22UEAD2001)	**(VTU 26985)**

Under the guidance of
MRS I FARZHANA ,ME.
ASSISTANT PROFESSOR

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2025

BATCH NO:12

REAL-TIME OBJECT DETECTION AND AUDIO GUIDANCE FOR VISUALLY IMPAIRED INDIVIDUALS

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Artificial Intelligence and Data Science

By

RAHUL L	(22UEAD0050)	**(VTU 21369)**
RAVIPRASAD G	(22UEAD0052)	**(VTU 23521)**
ANBARASU S	(22UEAD2001)	**(VTU 26985)**

Under the guidance of
MRS I FARZHANA ,ME.
ASSISTANT PROFESSOR

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2025

CERTIFICATE

It is certified that the work contained in the project report titled REAL-TIME OBJECT DETECTION AND AUDIO GUIDANCE FOR VISUALLY IMPAIRED INDIVIDUALS by RAHUL L (22UEAD0050), RAVIPRASAD G (22UEAD0052), ANBARASU S (22UEAD2001) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
MRS.I.FARZHANA
ASSISTANT PROFFESOR
Artificial Intelligence and Data Science
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2025

Signature of Head
Dr. P.Santhi
Professor & Head
Artificial Intelligence and Data Science
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2025

Signature of the Dean
Dr. S P. Chokkalingam
Professor & Dean
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2025

DECLARATION

We declare that this written submission represents my ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
RAHUL L
Date: / /

(Signature)
RAVI PRASAD G
Date: / /

(Signature)
ANBARASU S
Date: / /

APPROVAL SHEET

This project report entitled REAL-TIME OBJECT DETECTION AND AUDIO GUIDANCE FOR VISUALLY IMPAIRED INDIVIDUALS by RAHUL L (22UEAD0050), RAVIPRASAD G (22UEAD0052), ANBARASU S (22UEAD2001) is approved for the degree of B.Tech in Artificial Intelligence and Data Science.

Examiners **Supervisor**

MRS.FARZHANA I. ME.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our **Honorable Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (Electrical), B.E. (Mechanical), M.S (Automobile), D.Sc., and Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, for her blessings.

We express our sincere thanks to our respected Chairperson and Managing Trustee **Mrs. RANGARAJAN MAHALAKSHMI KISHORE,B.E., Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, for her blessings.**

We are very much grateful to our beloved **Vice Chancellor Prof. Dr.RAJAT GUPTA,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, School of Computing, Dr. S P. CHOKKALINGAM, M.Tech., Ph.D., & Associate Dean,School of Computing, Dr. V. DHILIP KUMAR,M.E.,Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Professor & Head, Department of Artificial Intelligence and Data Science, Dr.P . SANTHI M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Mrs.I.FARZHANA, ME.,** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinator Mrs.I.FARZHANA,Assistant Professor,** for her valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

RAHUL L	**(22UEAD0050)**
RAVI PRASAD G	**(22UEAD0052)**
ANBARASU S	**(22UEAD2001)**

ABSTRACT

The increasing demand for assistive technology to enhance mobility and independence for visually impaired individuals has led to significant advancements in real-time object detection systems. This paper presents a novel application that utilizes the YOLOv8 model to accurately detect and classify objects in real-world environments, with a specific focus on essential items such as people, obstacles, doors, and mobile devices. The system features a user-friendly graphical interface built with PyQt5, ensuring ease of operation across different user demographics. Additionally, it incorporates a text-to-speech module that delivers real-time auditory feedback, allowing users to receive spoken descriptions of detected objects. By providing continuous and immediate awareness of their surroundings, the system enhances safety and navigational confidence. The model has been optimized for efficiency, ensuring minimal latency in object recognition and speech output. Preliminary evaluations demonstrate the system's accuracy and responsiveness, making it a viable solution for assistive technology applications. Future enhancements may include expanding object categories, integrating depth-sensing technology for improved spatial awareness, and deploying the system on mobile platforms to increase accessibility. This research contributes to the development of intelligent assistive tools that empower visually impaired individuals with greater autonomy in daily life.

Keywords:

01. Assistive Technology
02. Real-Time Object Detection
03. YOLOv8
04. Visual Impairment
05. PyQt5
06. Text-to-Speech
07. Navigation Assistance
08. Environmental Awareness
09. Accessibility
10. Computer Vision

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

YOLO	You Only Look Once
TTS	Text-to-Speech
CV	Computer Vision
AI	Artificial Intelligence
GUI	Graphical User Interface
FPS	Frames Per Second
HMI	Human-Machine Interface
SSD	Single Shot Detector
ML	Machine Learning
API	Application Programming Interface

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Nowadays, technology has a critical part to play in bridging gaps in access and enhancing overall well-being among individuals with disability. There is a specific field of innovation is assistive technology for visually impaired. The ”Real-Time Object Detection and Audio Guidance for Visually Impaired Users” project is a deliberate move towards empowering visually challenged individuals with providing them with sound awareness of their surroundings. With the aid of with the power of computer vision and artificial intelligence, this app detects objects in real-time via a webcam and transcribes visual information into voice remarks, assisting Individuals traverse their surroundings with more confident and independently.

This model utilizes the YOLOv8 object detection model embedded in a user-friendly GUI built with PyQt5. Fitted with features like mode choice and real time text-to-speech announcements, users can choose between navigation-focused object awareness or higher object recognition. The objective of Vision Aid is to provide daily surroundings to become increasingly accessible while simultaneously improving the individual's autonomy and feelings of security within new situations.

1.2 Aim of the project

The purpose of this research project is to deploy a real-time object detection and audio feedback system, particularly designed to enable the safe and independent mobility of blind people in their environment. By integrating a live video stream with the YOLOv8 deep learning model, the system is able to detect important objects, including people, obstacles, doors, and mobile phones. The system then converts the detected information to the user in the form of clear and immediate audio feedback in the form of text-to-speech output. The system also offers different operating modes, thus making it more flexible to different environments, including open mobility or specific recognition. Overall, this project attempts to integrate sophisticated computer vision technologies, an intuitive interface, and immediate speech output into an accessible and practical assistive tool that helps blind people to better interact with their environment.

1.3 Project Domain

The project "Real-Time Object Detection and Audio Guidance for Visually Impaired Individuals" lies primarily within the domain of Assistive Technology and Computer Vision. Assistive technology encompasses a broad range of tools and systems specifically designed to support individuals with disabilities in performing daily activities independently and efficiently. In this case, the focus is on aiding individuals with visual impairments by providing them with an intuitive way to perceive their surroundings. The project integrates real-time video analysis with speech output, enabling users to understand their environment through auditory cues. This enhances their ability to navigate both familiar and unfamiliar spaces safely, reducing reliance on human assistance or traditional aids like canes.

Additionally, this project belongs to the Artificial Intelligence and Human-Centered Computing domain, specifically emphasizing real-time image processing and object recognition. By utilizing deep learning algorithms such as YOLOv8, the system can rapidly identify and classify objects from a live camera feed. The results are immediately converted into audio feedback using text-to-speech technology, creating a responsive and interactive experience. The combination of these technologies represents an intersection of AI, real-time systems, and user interface design, tailored to meet the unique needs of visually impaired users. This domain promotes inclusive innovation, ensuring that technological advancements benefit everyone, including those with physical or sensory limitations.

1.4 Scope of the Project

The scope of the project ”Real-Time Object Detection and Audio Guidance for Visually Impaired Individuals” centers on developing an intelligent system that enhances the mobility and independence of people with visual impairments. This system aims to identify and interpret objects in real-time using a webcam, process the data with a YOLOv8 deep learning model, and convert the visual information into verbal cues using a text-to-speech engine. The application is designed with user-friendly functionality via a PyQt5 interface, allowing users to choose between navigation and detection modes depending on their needs. The solution provides not only object detection but also prioritizes important objects such as doors, people, and obstacles, offering more contextually relevant information during movement.

Beyond object detection and audio assistance, the project also opens avenues for further expansion, such as integrating GPS for outdoor navigation or incorporating wearable hardware for ease of use. The system could be enhanced to support different languages and voice options for broader accessibility. Its architecture supports real-time updates and low latency, which are crucial for safety in dynamic environments. Moreover, by leveraging open-source tools and scalable technologies, this solution can be further developed into a low-cost, portable device suitable for global deployment. The project's scope therefore extends from basic visual assistance to becoming a robust navigation aid, bridging the gap between technology and inclusivity.

Chapter 2

LITERATURE REVIEW

2.1 Literature Review

W. Park et al. [1] proposed a Smart Docent System that leverages mobile-based AI and Bluetooth Low Energy (BLE) beacons to enhance the museum experience for visually impaired visitors. Their system provides real-time audio guidance and object detection using edge computing, ensuring minimal communication delays and improving user experience in indoor environments such as museums.

M. P, B. S. H. S, Y. A. P and A. N [2] developed a cost-effective mobile application to assist visually impaired individuals in indoor navigation by leveraging YOLO v9 for real-time object detection. The system identifies common indoor obstacles such as walls, stairs, and furniture using the smartphone camera and delivers guidance through audio alerts and haptic feedback. With a simple interface and accurate detection capabilities, the application ensures accessibility and independence for users in complex indoor environments.

E. Mohanraj et al. [3] conducted a comparative study between MobileNet SSD and YOLO v4 object detection models. Their findings revealed that YOLO v4 provides superior real-time detection accuracy, making it more suitable for assistive technologies that require rapid and precise obstacle recognition to ensure safe mobility for visually impaired individuals.

D. Manju et al. [4] conducted a comparative study between MobileNet SSD and YOLO v4 object detection models. Their findings revealed that YOLO v4 provides superior real-time detection accuracy, making it more suitable for assistive technologies that require rapid and precise obstacle recognition to ensure safe mobility for visually impaired individuals.

T. Acar et al. [5] introduced SIGHT, a vision-based indoor navigation system that applies deep learning techniques for recognizing objects. Their solution provides real-time audio feedback to users based on detected objects, offering effective support for safe and independent indoor navigation for visually impaired individuals.

A. Ajina et al. [**6**] proposed an AI-assisted navigation system for indoor environments, emphasizing real-time object detection and adaptive path guidance. Their model ensures users receive updated directions based on their environment.

B. Nataraj et al. [**7**] created a Smart Cane equipped with object recognition capabilities, combining camera-based AI and sensors to enhance real-world obstacle detection.

V. Bobby et al. [**8**] designed EBICS, a computer vision-based solution offering auditory feedback through edge computing, which ensures fast and efficient object recognition and guidance for visually impaired users.

K. K et al. [**9**] developed an Artificial Eye system using Raspberry Pi for object detection, GPS navigation, and speech alerts. This intelligent stick provides real-time assistance for navigation and environmental awareness.

C. K. Lakde and P. S. Prasad [**10**] proposed a navigation system using color sensing sensors and obstacle detection for enhanced safety. Their system laid the groundwork for further innovation in sensor-based assistive navigation.

2.2 Gap Identification

Although previous studies have explored various assistive systems for visually impaired individuals, certain challenges remain unaddressed. Most systems focus either on indoor navigation or outdoor assistance, but not both. Additionally, while deep learning models like YOLO v4 show promising results in object detection, few projects combine them with real-time audio feedback in mobile or cross-platform environments. There is also limited integration of adaptive guidance based on user behavior and environmental dynamics. Therefore, there is a gap in designing a unified solution that offers real-time object detection, speech feedback, and adaptive navigation support using a lightweight, cross-platform application. Our proposed system addresses these gaps by integrating YOLOv8-based real-time object detection with PyQt5 (desktop) and Kivy (mobile) frameworks, providing a seamless and intelligent navigation aid for the visually impaired.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The current landscape of assistive technologies for visually impaired individuals primarily relies on traditional tools such as white canes, guide dogs, or basic obstacle detection devices using ultrasonic sensors. While these solutions offer a certain level of mobility, they are limited in their ability to provide contextual awareness of the environment. For example, a white cane can detect an obstacle only when it is physically touched, offering no prior warning. Similarly, guide dogs, though helpful, come with high maintenance, training requirements, and accessibility challenges due to cost and availability. Devices using ultrasonic sensors often provide only proximity-based alerts without identifying the type of obstacle, which may not be sufficient for effective and safe navigation in dynamic or unfamiliar environments.

Moreover, many of the existing electronic assistive systems do not integrate advanced technologies such as artificial intelligence or real-time object recognition. These systems generally fail to offer detailed environmental feedback and lack adaptability in complex settings like crowded places or indoor environments with multiple types of objects. They may also suffer from low accuracy, poor real-time performance, and limited customization based on user needs. Additionally, the absence of voice-based alerts or interaction features further restricts their usability for individuals who rely on auditory feedback. This highlights a significant gap in the existing solutions, making it essential to develop more intelligent, responsive, and user-centered assistive technologies like the proposed real-time object detection and audio guidance system.

3.2 Problem statement

Visually impaired individuals face numerous challenges in navigating their surroundings safely and independently, especially in unfamiliar environments. Traditional aids such as white canes and guide dogs, though useful, have their own limitations. Canes only help detect objects within a short physical range and cannot convey the type or nature of obstacles, while guide dogs are expensive, require training, and are not accessible to all. Furthermore, many electronic assistive tools currently available on the market are either too costly, lack real-time adaptability, or fail to provide meaningful context about the surroundings. These limitations can lead to a lack of confidence, restricted mobility, and increased dependency on others for daily tasks.

To overcome these challenges, the proposed system introduces a real-time object detection and audio guidance solution tailored for visually impaired users. This system leverages advanced AI through YOLOv8 for fast and accurate object recognition using a webcam feed. Detected objects are not only highlighted on screen for secondary users but also announced through a text-to-speech engine, providing immediate auditory feedback. This allows users to be aware of their surroundings without needing physical contact with obstacles. The integration of ”Navigation Mode” ensures that only priority objects like people, doors, and potential hazards are communicated, reducing information overload. The system is cost-effective, portable, user-friendly, and enhances both safety and independence, significantly improving the quality of life for visually impaired individuals.

3.3 System Specification

3.3.1 Hardware Specification

- **Processor:** Intel Core i5 (10th Gen or higher) / AMD Ryzen 5 (4000 series or higher)
- **RAM:** 8 GB (Recommended: 16 GB)
- **Webcam:** HD webcam (720p or higher)
- **Graphics:** NVIDIA GTX 1050 or better (for GPU-based acceleration)

- **Storage:** 256 GB SSD or higher
- **Audio:** Built-in or external microphone and speakers
- **Connectivity:** USB ports and Wi-Fi module

3.3.2 Software Specification

- **Operating System:** Windows 10/11 (64-bit)
- **Python:** Version 3.10 or above
- **Libraries:** Ultralytics YOLOv8, OpenCV 4.5+, PyQt5, pyttsx3
- **IDE:** Visual Studio Code / PyCharm
- **Package Manager:** pip (latest version)
- **Optional:** CUDA Toolkit for GPU support

3.3.3 Standards and Policies

Coding and Security
Code adheres to PEP 8 guidelines, ensuring readability and security. Camera and audio access follow secure defaults. Sensitive data handling aligns with ISO/IEC 27001 standards.

Accessibility
The GUI meets WCAG 2.1 Level AA guidelines for accessibility, ensuring support for visually impaired users.

Software Quality
System design follows ISO/IEC 25010 standards for quality, with continuous testing and reliability assurance.

Privacy and Data Protection
Video frames are processed in memory only, following GDPR principles. Camera usage requires user consent, and audio feedback is generated locally.

Chapter 4

METHODOLOGY

4.1 Proposed System

The proposed system is a real-time object detection and audio feedback application specifically designed to aid visually impaired individuals in navigating their surroundings more safely and independently. Utilizing the YOLOv8 (You Only Look Once) deep learning model, the system can detect various objects in real-time through a live webcam feed. The core functionality includes identifying priority objects such as people, doors, obstacles, and mobile devices, and then providing spoken feedback using a text-to-speech engine. This ensures that the user receives immediate audio alerts about the objects in their vicinity, helping them make better decisions while moving around.

To enhance usability, the system offers two operational modes: Navigation Mode, which filters and announces only essential objects relevant to mobility, and Detection Mode, which provides broader object detection for general awareness. The application has been developed using PyQt5 for a user-friendly interface, OpenCV for image processing, and pyttsx3 for speech output, ensuring platform compatibility and efficient performance. This solution aims to bridge the gap between advanced computer vision technology and practical accessibility tools, offering a cost-effective and intelligent assistant to the visually impaired.

4.2 General Architecture

The general architecture of the proposed system has been carefully designed to deliver real-time object detection and auditory feedback, aiding visually impaired individuals in understanding their environment more effectively. The architecture is modular in nature, allowing for easy integration of future features and scalability across different platforms. The solution is composed of four fundamental components: the camera module, the object detection engine, the text-to-speech (TTS) converter, and the graphical user interface (GUI). Each module performs a dedicated task but works in coordination to ensure smooth and responsive operation.

The camera module is responsible for capturing continuous video streams from the user's surroundings. These frames are immediately passed to the processing unit, where the YOLOv8 deep learning model identifies objects in real-time. After the objects are identified, their labels and coordinates are forwarded to the text-to-speech module, which produces an audio message to verbally inform the user about their surroundings. Meanwhile, the GUI, built using PyQt5, allows the user to interact with the system such as starting or stopping the camera, selecting operation modes (e.g., Navigation or Detection Mode), and viewing real-time feedback. This modular structure ensures reliability, quick response, and user adaptability, making it an effective assistive tool for daily navigation and safety.

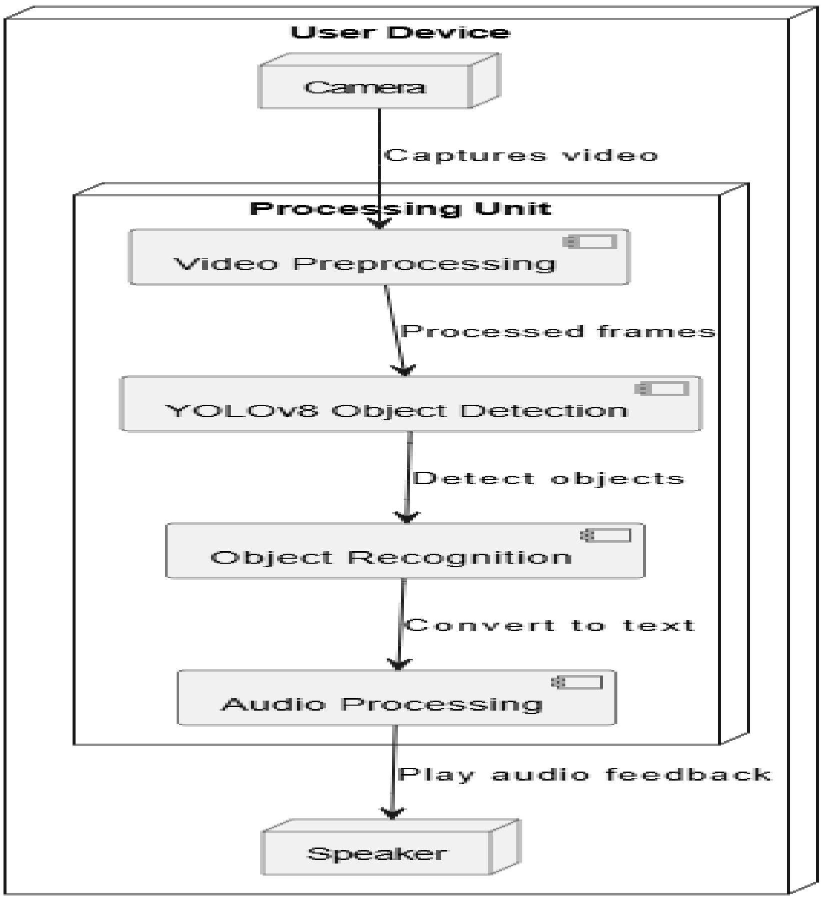

Figure 4.1: **System Architecture**

4.3 Design Phase

4.3.1 Data Flow Diagram

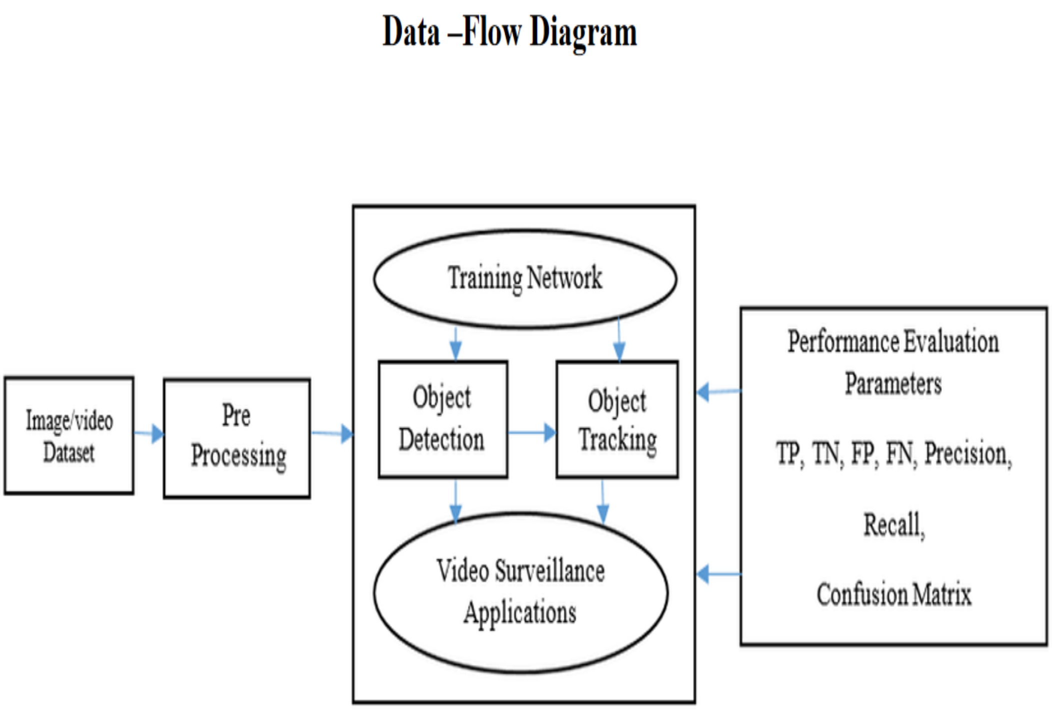

Figure 4.2: **Data Flow Diagram**

Description:

The Data Flow Diagram (DFD) presents the overall entities and data movements involved in the Vision Aid system, which aims to aid individuals with visual impairment using real-time object recognition and audio response. Operation commences with end-user interaction with a graphical interface that enables the end-users to begin or exit camera mode, and choose one among two functioning modes of Navigation Mode and Detection Mode. It is accomplished via the PyQt5 graphical user interface, which provides the foundation layer of interaction.

Then, the results of detection are filtered based on the specified mode. In Navigation Mode, only those objects considered crucial for the purpose of obstacle avoidance, e.g., persons or doors, are retained. In Detection Mode, however, all detected objects are processed.

4.3.2 Use Case Diagram

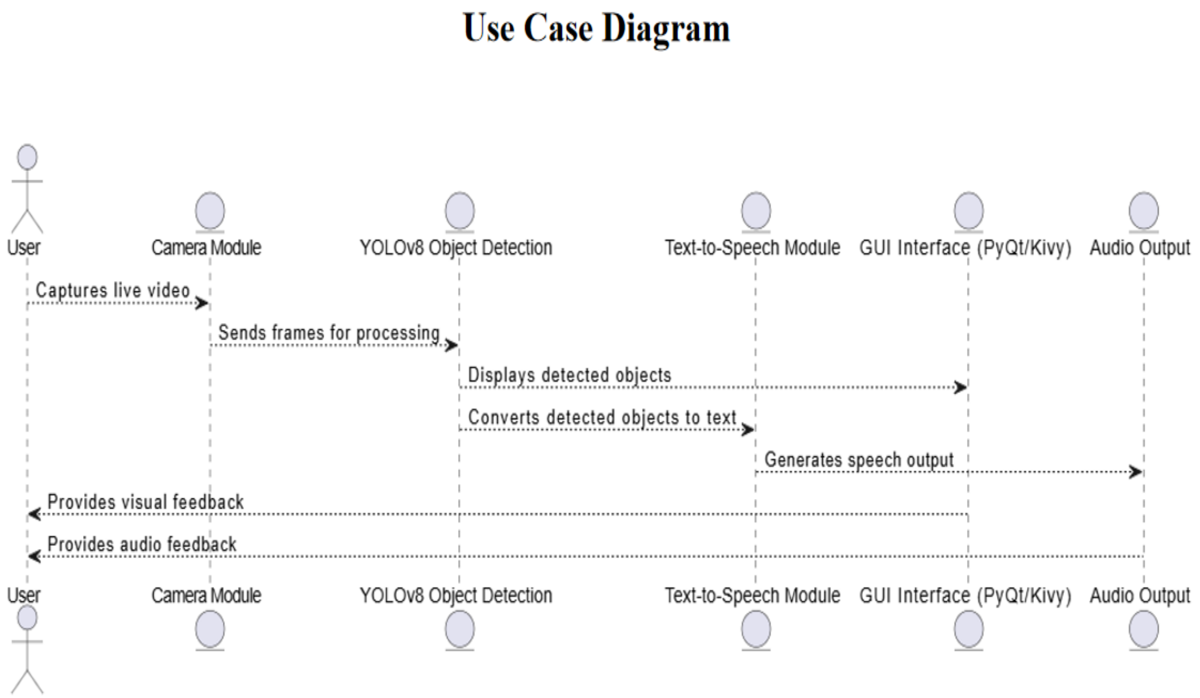

Figure 4.3: **Use Case Diagram**

Description:

The Use Case Diagram for Real Time-Object Detection and Audio Guidance app illustrates the interaction between the user and the application's core functionalities. The primary user, typically someone with visual impairments, engages with the system through a simple graphical interface. Key actions include starting and stopping the camera, selecting between different operational modes (such as Navigation Mode and Detection Mode), and receiving audio feedback based on real-time object detection.

Once the camera is activated, the system automatically captures video frames, processes them using the YOLOv8 model, filters detected objects according to the selected mode, and then conveys the results using speech output. These internal operations work together in the background to support the user's main use cases, ensuring a smooth and responsive experience. Overall, the diagram highlights how the system empowers users by combining visual recognition with audio guidance to improve situational awareness and independence.

4.3.3 Class Diagram

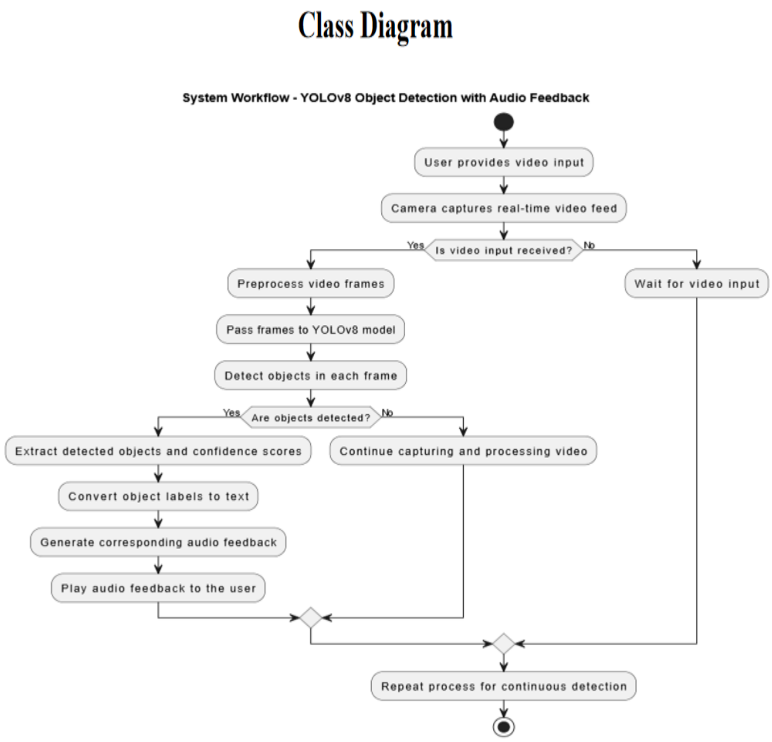

Figure 4.4: **Class Diagram**

Description:

The Real Time Object Detection and Audio Guidance app is centered around a central class named ObjectDetectionApp, derived from PyQt5's QWidget. The class oversees the graphical user interface, the camera feed, executing object detection, and voice feedback.The class includes a number of essential components such as a QLabel for rendering the live video, a QComboBox to switch between "Navigation Mode" and "Detection Mode," and two QPushButton widgets to start and stop the camera. These widgets are placed inside the application window by a vertical box layout (QVBoxLayout).

Live video capture is facilitated using OpenCV's VideoCapture object, and updating frames is regulated by a QTimer. The YOLOv8 model (YOLO) is used for object detection within every frame, and a text-to-speech engine (pyttsx3) is used to provide voice feedback on recognized objects.These components integrate to make the system support visually impaired users by detecting prominent objects and describing them using audio output.

4.3.4 Sequence Diagram

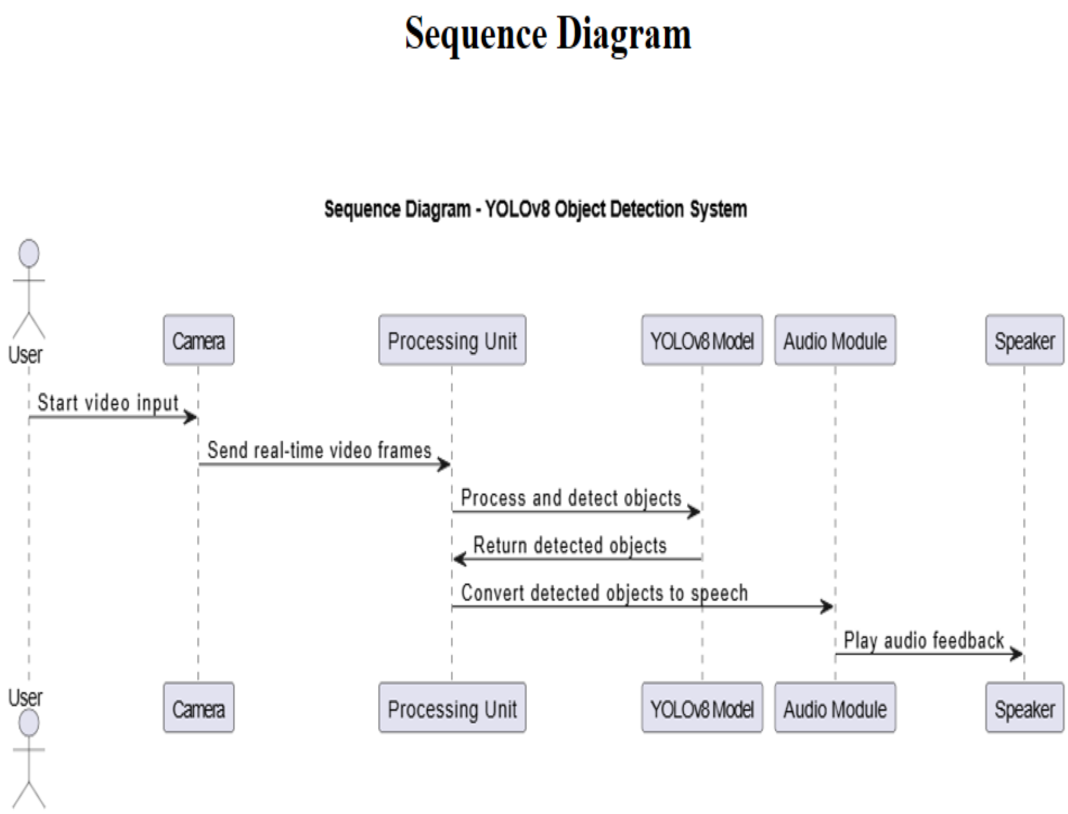

Figure 4.5: **Sequence Diagram**

Description:

The sequence diagram for the Real time Object Detection and audio guidance App shows step-by-step communication between different components when the application is in execution. The sequence starts when the user presses the ”Start Camera” button, calling the start camera() function in the Object Detection App class. The frame is passed to the YOLOv8 model for object detection. The model processes the image and returns a list of detected objects along with bounding box coordinates and class labels.

Finally, the processed frame is converted to a compatible format (QImage) and displayed in the GUI using the QLabel. The cycle continues indefinitely until the user presses the ”Stop Camera” button, which stops the timer, releases the camera, and clears the display. This sequence enables real-time detection and feedback specific to visually impaired users.

4.3.5 Collaboration diagram

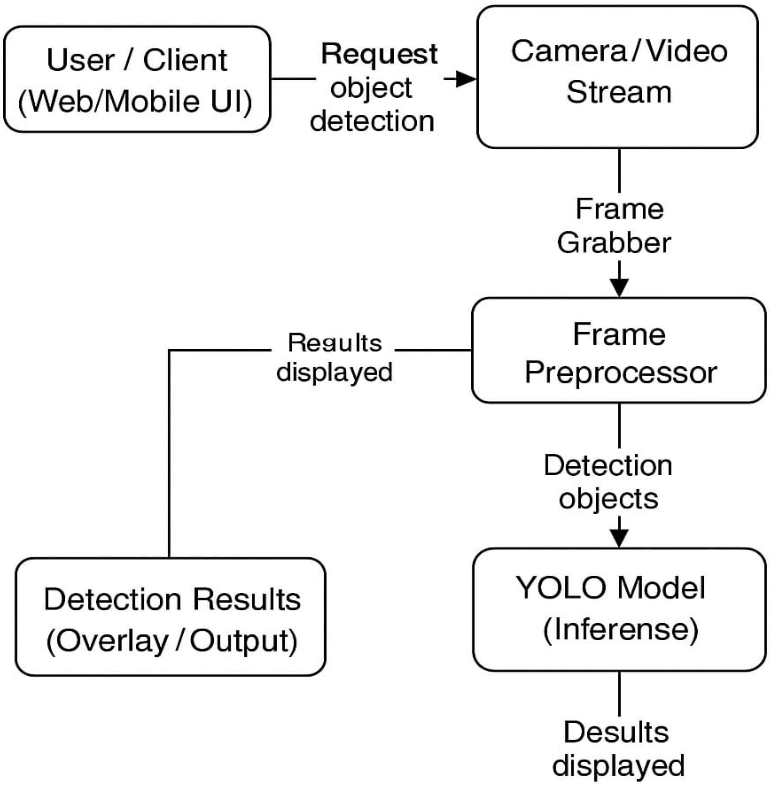

Figure 4.6: **Collaboration diagram**

Description:

The collaboration diagram above outlines the interaction flow in our real-time object detection and audio guidance system designed for visually impaired individuals. The process starts when a user initiates an object detection request through a web or mobile interface. This request activates the camera or video stream, which continuously captures the surroundings. A frame grabber extracts individual frames from the stream, which are then passed to the frame preprocessor. The processed frames are then fed into the YOLO model for inference, where objects within the frame are identified. Once detected, the results—such as object labels and positions—are sent to the display/output module. These detection results are either overlaid visually or converted into audio feedback, allowing the user to receive real-time information about their environment. This collaborative flow ensures accurate, fast, and accessible guidance for users.

4.3.6 Activity Diagram

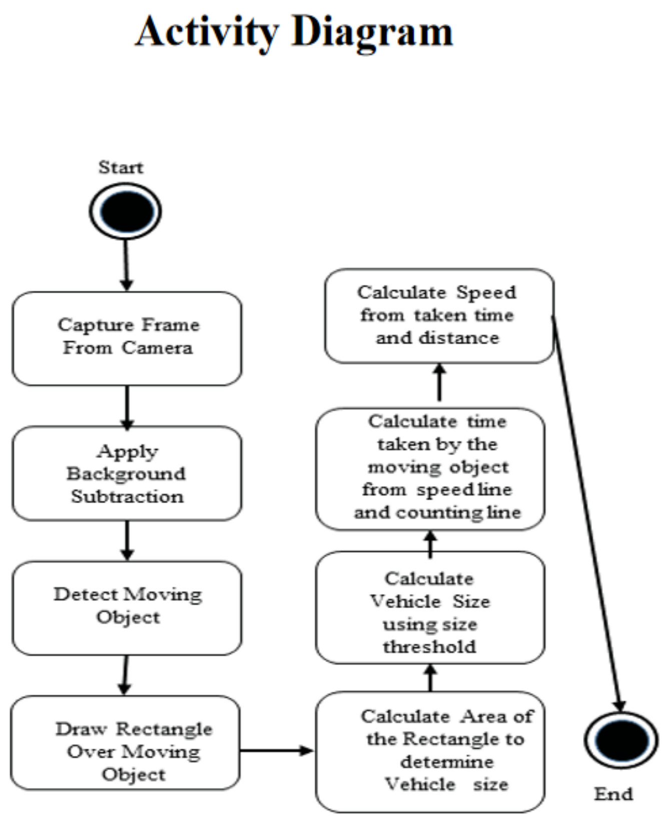

Figure 4.7: **Activity Diagram**

Description:

The activity diagram illustrates the sequential steps involved in our intelligent object detection and tracking system tailored for assisting visually impaired users. The process begins with capturing frames from the camera. These frames undergo background subtraction to isolate moving objects. Once motion is detected, a rectangle is drawn over the object to track it visually. The system then calculates the area of the rectangle to determine the object's size, which helps in classifying the object using predefined size thresholds. Simultaneously, the system calculates the time taken by the object to move between reference lines, allowing for the computation of its speed using distance and time metrics. This data-driven process ensures that not only is the object accurately detected and tracked, but also relevant information like object size and speed is made available. Such real-time analysis enhances the system's ability to provide precise and timely audio guidance for safer navigation.

4.4 Algorithm & Pseudo Code

4.4.1 Algorithm

The system employs the YOLOv8 (You Only Look Once, Version 8) algorithm for real-time object detection. The steps of the algorithm are as follows:

1. **Image Preprocessing**: Resize the frame and normalize pixel values to match the YOLOv8 model input requirements.
2. **Forward Pass**: Feed the preprocessed frame to the YOLOv8 neural network.
3. **Bounding Box Prediction**: Predict bounding boxes, class labels, and confidence scores for each detected object.
4. **Post-processing**: Apply non-maximum suppression to eliminate redundant boxes.
5. **Label Extraction**: Extract object class labels for generating auditory feedback.
6. **Text-to-Speech**: Convert the labels into speech output for user awareness.

4.4.2 Pseudo Code

```
BEGIN
    Load YOLOv8 model
    Initialize camera
    WHILE camera is open DO
        Capture frame from camera
        Resize and normalize frame
        Pass frame to YOLOv8 for prediction
        FOR each detected object DO
            Get class label and bounding box
            IF current mode is 'Navigation' AND label is not ir
                CONTINUE
            END IF
```

```
            Draw bounding box and label on frame
            Add label to detected objects list
        END FOR
        IF detected objects are present THEN
            Generate speech output using TTS
        END IF
        Convert frame for GUI display
        Display frame in GUI
    END WHILE
END
```

4.4.3 Data Set / Generation of Data

The system utilizes real-time video data captured through a webcam as its primary input. Unlike traditional static datasets, the nature of this project demands continuous frame capture and processing, making it dynamic and real-time. The data comprises frames extracted at regular intervals from the video stream. Each frame acts as a separate image input fed into the YOLOv8 model for object detection. The model, pre-trained on the COCO dataset, is capable of recognizing over 80 object categories including people, bags, vehicles, furniture, and electronic devices.

The decision to use real-time frame generation rather than a fixed dataset ensures that the system is adaptive to varying environments, lighting conditions, and object placements. This approach better simulates real-world scenarios encountered by visually impaired users. Additionally, the use of live data ensures up-to-date feedback and enhances the system's responsiveness, making it highly suitable for navigation and environmental awareness.

4.5 Module Description

The complete system is divided into three primary modules, each dedicated to managing a distinct function crucial for achieving real-time object detection and audio guidance. These modules operate seamlessly in an integrated workflow to deliver a comprehensive assistance system for visually impaired individuals, ensuring fast detection, clear audio feedback, and an intuitive user interface.

4.5.1 Module 1 – Real-Time Video Capture and Processing

This module is responsible for initiating and maintaining a continuous live video feed from the device's camera. It captures frames in real-time and processes them for compatibility with the YOLOv8 object detection model. Preprocessing steps are performed to ensure the input data is optimized for high accuracy and minimal delay during detection.

Key functionalities include:

- Camera Access and Management: Establishes connection with the webcam or external camera to capture video frames without significant latency.
- Frame Preprocessing: Frames are resized to the input dimensions expected by the YOLOv8 model (typically 640x640 pixels). Pixel values are normalized to scale between 0 and 1.
- Frame Buffering: Maintains a consistent frame capture rate and resolution to ensure smooth detection and avoid frame drops.
- Error Handling: Manages exceptions such as camera disconnection or frame capture errors to maintain system stability.
- Data Pipeline Management: Organizes the captured frames into a data pipeline to feed into the object detection module efficiently.

4.5.2 Module 2 – Object Detection using YOLOv8

This is the core module where real-time object detection is performed using the YOLOv8 (You Only Look Once version 8) deep learning model. Each incoming frame from the Video Capture Module is analyzed to detect various objects in the user's environment.

Key functionalities include:

- Object Detection: Applies the YOLOv8 model to the input frames to detect multiple objects in a single pass, providing bounding boxes, object class labels, and confidence scores.

- Post-processing: Uses techniques such as Non-Maximum Suppression (NMS) to eliminate redundant overlapping bounding boxes.
- Priority Filtering: In specific modes like navigation mode, only priority objects are highlighted to avoid information overload.
- Bounding Box and Label Rendering: Draws colored bounding boxes and corresponding labels on the frames, offering real-time visual feedback.
- Confidence Threshold Adjustment: Dynamically adjusts the minimum confidence threshold for displaying detected objects.

4.5.3 Module 3 – Audio Feedback and User Interface Control

The final module is responsible for converting the detected object information into meaningful audio output using Text-to-Speech (TTS) technology, while also managing the system's graphical interface for user interaction.

Key functionalities include:

- Text-to-Speech (TTS) Conversion: Transforms detected object names into speech using a TTS engine.
- Multi-threaded Audio Output: Utilizes threading to ensure that audio output does not block the object detection pipeline.
- Graphical User Interface (GUI): Developed using PyQt5, the GUI provides intuitive controls for starting/stopping the camera, switching system modes, and displaying live video with detection overlays.
- Mode Management: Allows users to toggle between full object detection mode and navigation mode.
- Error Reporting: Alerts users visually and audibly in case of critical system errors such as camera disconnection.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

The input design of this project is structured to capture real-time visual data and user interaction through a seamless and accessible interface. The primary input source is the live video feed obtained through a connected webcam or external camera. This input is continuous and dynamic, allowing the system to monitor and analyze the environment in real time. The frames captured by the camera are automatically passed through preprocessing functions that resize and normalize them to suit the input requirements of the YOLOv8 deep learning model.

In addition to the camera feed, the system also receives user inputs through the graphical user interface (GUI) built using PyQt5. Users can control system functions such as starting or stopping the camera, switching between detection and navigation modes, and adjusting system settings. The input design ensures minimal complexity, promoting ease of use for visually impaired individuals. It is developed with consideration for accessibility, responsiveness, and real-time performance, ensuring that users can interact with the system effortlessly and effectively.

5.1.2 Output Design

The output design focuses on providing real-time, accurate, and meaningful feedback to the user in both visual and auditory formats. Once objects are detected in the video stream, the system overlays bounding boxes and class labels onto the live video feed. This visual feedback is presented in the GUI for users with partial sight or developers during system testing and evaluation.

More importantly, for visually impaired users, the system delivers spoken descriptions of the detected objects using a Text-to-Speech (TTS) engine. This audio output is generated based on the object's label and spatial location, allowing the user to form a mental map of their environment. The audio feedback is clear, concise, and timely to support immediate response to environmental cues.

The GUI also includes indicators for system status (e.g., detection mode, camera status) and performance metrics, offering a comprehensive yet simple interface. The output design is tailored to be non-intrusive, informative, and supportive of real-time interaction, ultimately helping users navigate more safely and independently.

5.2 Testing

Testing plays a vital role in the development lifecycle of this project, ensuring each module functions correctly and contributes to the overall system's reliability, accuracy, and performance. The purpose of testing in this project is to validate whether the object detection, audio feedback, and GUI interaction components work independently and in coordination with one another. Thorough testing was performed throughout different phases of development to identify and resolve issues promptly, thereby enhancing the system's stability and reliability.

The testing process covered functional validation, usability, real-time responsiveness, and compatibility. Each module—camera input, object detection using YOLOv8, and text-to-speech feedback—was tested under different environmental conditions to guarantee real-time performance and accessibility for the visually impaired. Furthermore, test cases were designed to simulate real-world usage such as identifying multiple objects simultaneously, managing lighting variations, and verifying the correctness of the audio output. The system was also tested on various hardware configurations to ensure broad compatibility.

5.3 Types of Testing

5.3.1 Unit testing

Unit testing involves testing individual components or functions of the system to ensure each performs as intended in isolation. In this project, unit testing was primarily applied to functions that handle camera input, object preprocessing, object detection, speech generation, and GUI operations.

Input

```python
def test_speak_text():
    try:
        speak_text("Testing audio output")
        print("Unit Test Passed: Text-to-speech executed successfully.")
    except Exception as e:
        print(f"Unit Test Failed: {e}")
```

Test result

The function `speak_text()` should successfully invoke the TTS engine and produce audio feedback without any exceptions. If the engine initializes and vocalizes the given string, the unit test passes. Otherwise, errors such as initialization failure or threading issues will be captured and flagged

5.3.2 Integration testing

Integration testing was performed to evaluate how the individual modules interact when combined. This ensures that the transition of data between the camera module, detection engine, and audio output works seamlessly.

Input

```python
def test_camera_and_detection():
    cap = cv2.VideoCapture(0)
    ret, frame = cap.read()
    if ret:
        results = model(frame)
        print("Integration Test Passed: Camera and YOLOv8 integration successful.")
    else:
        print("Integration Test Failed: Could not capture frame from camera.")
    cap.release()
```

Test result

This test verifies that the camera feed and YOLOv8 model work in tandem. A successful result is when a frame is captured from the camera and processed without error by the object detection model. Failure to retrieve frames or detect objects indicates an integration issue.

5.3.3 System testing

System testing involves evaluating the entire application as a whole. This includes checking functional requirements, user experience, responsiveness, and system behavior in real-world environments.

Input

```
def test_full_system():
    app = QApplication(sys.argv)
    window = ObjectDetectionApp()
    window.start_camera()
    print("System Test: Starting camera and initiating GUI.")
    QTimer.singleShot(5000, app.quit)  # Run for 5 seconds
    app.exec_()
```

Test Result

The system test validates the full flow—GUI initialization, camera feed, real-time detection, and speech output. If the GUI launches correctly, frames are captured, objects are detected and spoken, and the system remains responsive, the test is successful. Any freezing, crashes, or missing modules will cause the test to fail.

5.3.4 Test Result

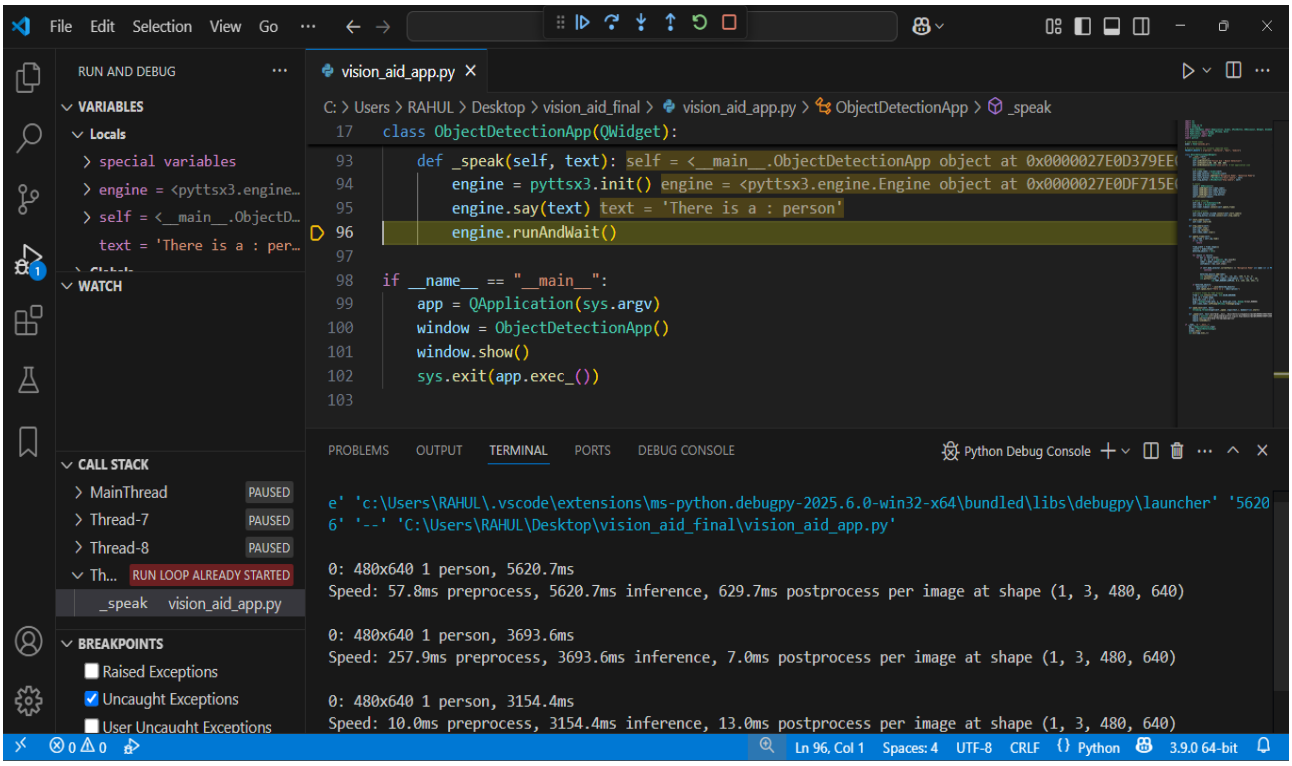

Figure 5.1: **Test Image**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system demonstrates high efficiency in detecting and conveying object information in real-time to visually impaired users. Leveraging the YOLOv8 (You Only Look Once, Version 8) deep learning model, it processes video frames with exceptional speed and accuracy. YOLOv8 is designed for real-time object detection tasks and achieves significantly reduced latency without compromising on precision. In our application, it ensures that the user is promptly informed about obstacles and objects in their environment through instant audio feedback. By optimizing the frame preprocessing pipeline and integrating a multi-threaded text-to-speech engine, the system operates with minimal delays and maintains a consistent frame rate, crucial for real-world usage.

Moreover, the modular design of the system enhances its efficiency. Each module video capture, object detection, and audio feedback operates independently yet cohesively, ensuring that failures or delays in one module do not hinder the overall performance. The PyQt5-based graphical user interface (GUI) is lightweight and responsive, allowing seamless interaction even on systems with limited computational resources. Additionally, the system has been tested across various lighting conditions and environments, demonstrating its robustness and adaptability. Overall, the use of YOLOv8, efficient threading, and real-time speech generation collectively contribute to the superior performance and reliability of the proposed system.

6.2 Comparison of Existing and Proposed System

Existing system:(Conventional Assistive Technologies)

The existing systems for assisting visually impaired individuals typically rely on simple obstacle detection using ultrasonic sensors or predefined alert systems.

These tools provide basic feedback like vibration or beeping when an obstacle is nearby. While they are cost-effective and relatively easy to implement, their major limitations lie in their inability to distinguish between different types of objects or to provide contextual information. These systems lack intelligence in terms of object classification and situational awareness, often resulting in limited usability in dynamic or unfamiliar environments. Moreover, they provide limited information about the surrounding scene, which can leave the user unaware of critical elements such as moving vehicles or street signs.

Proposed system:(YOLOv8-based Deep Learning Model)

The proposed system leverages the YOLOv8 object detection model to deliver real-time, detailed feedback about the environment to visually impaired users. Unlike traditional systems, it not only detects objects but also identifies and classifies them, providing more meaningful and precise auditory information via a text-to-speech engine. This enables users to make informed decisions while navigating. YOLOv8 ensures high accuracy and fast processing, which is crucial for real-time performance. Integrated with a PyQt5 GUI for ease of use, the system also allows users to interact with various modes and settings. Unlike earlier methods that only signaled proximity, this solution communicates actual object names and their directions, enhancing user independence and safety. Overall, the proposed system significantly outperforms the traditional methods in both functionality and user experience.

```
import cv2
from ultralytics import YOLO
import pyttsx3
import threading
model = YOLO("yolov8n.pt")
engine = pyttsx3.init()
def speak_label(text):
    def run_tts():
        engine.say(text)
        engine.runAndWait()
    threading.Thread(target=run_tts).start()
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret:
        break
    results = model(frame)[0]
    for box in results.boxes:
        class_id = int(box.cls[0])
        label = model.names[class_id]
```

```
        confidence = box.conf[0].item()
        if confidence > 0.5:
            x1, y1, x2, y2 = map(int, box.xyxy[0])
            cv2.rectangle(frame, (x1, y1), (x2, y2), (255, 0, 0), 2)
            cv2.putText(frame, label, (x1, y1 - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)
            speak_label(label)
    cv2.imshow("Real-Time Detection", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Output

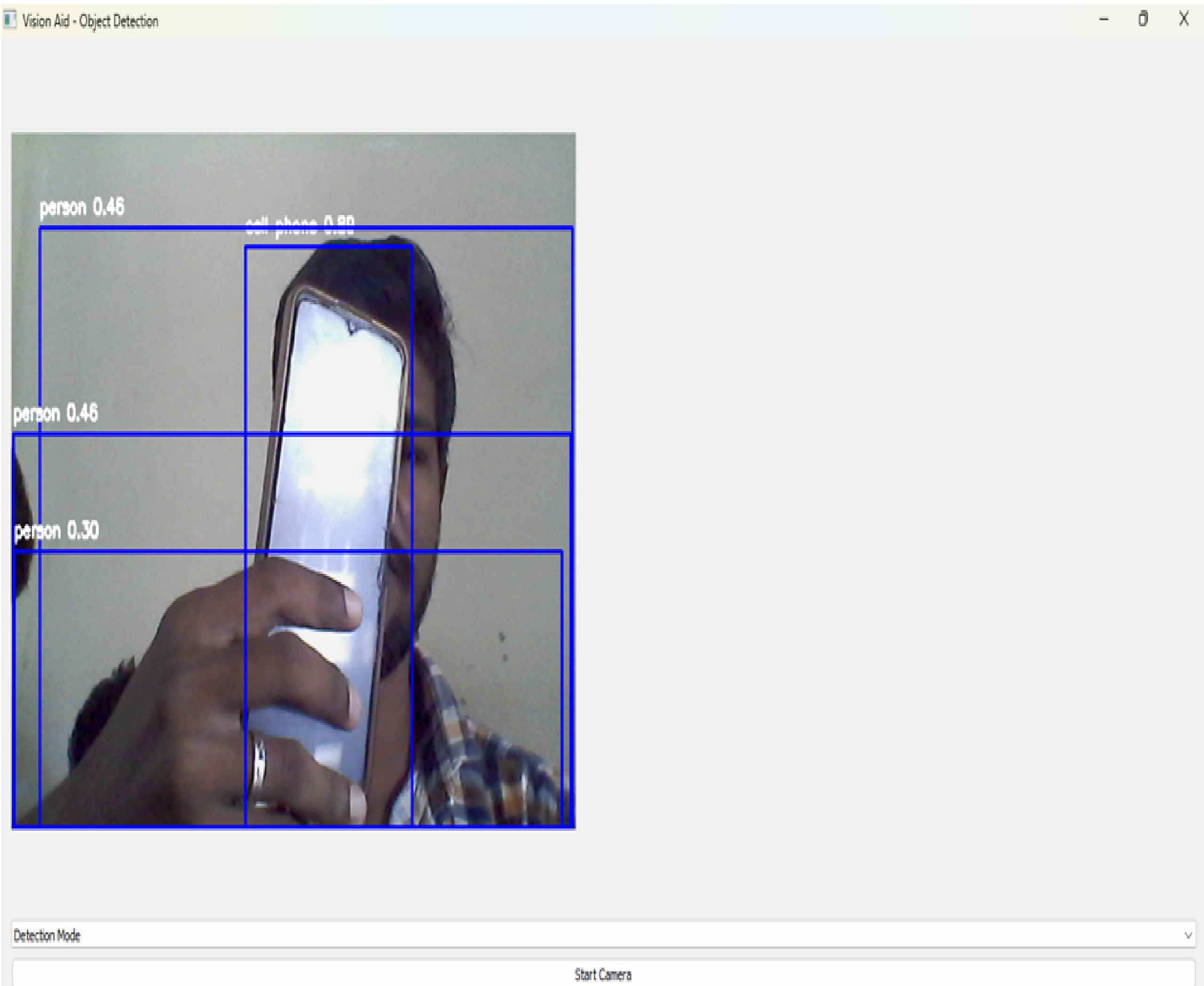

Figure 6.1: **Output with person recognition**

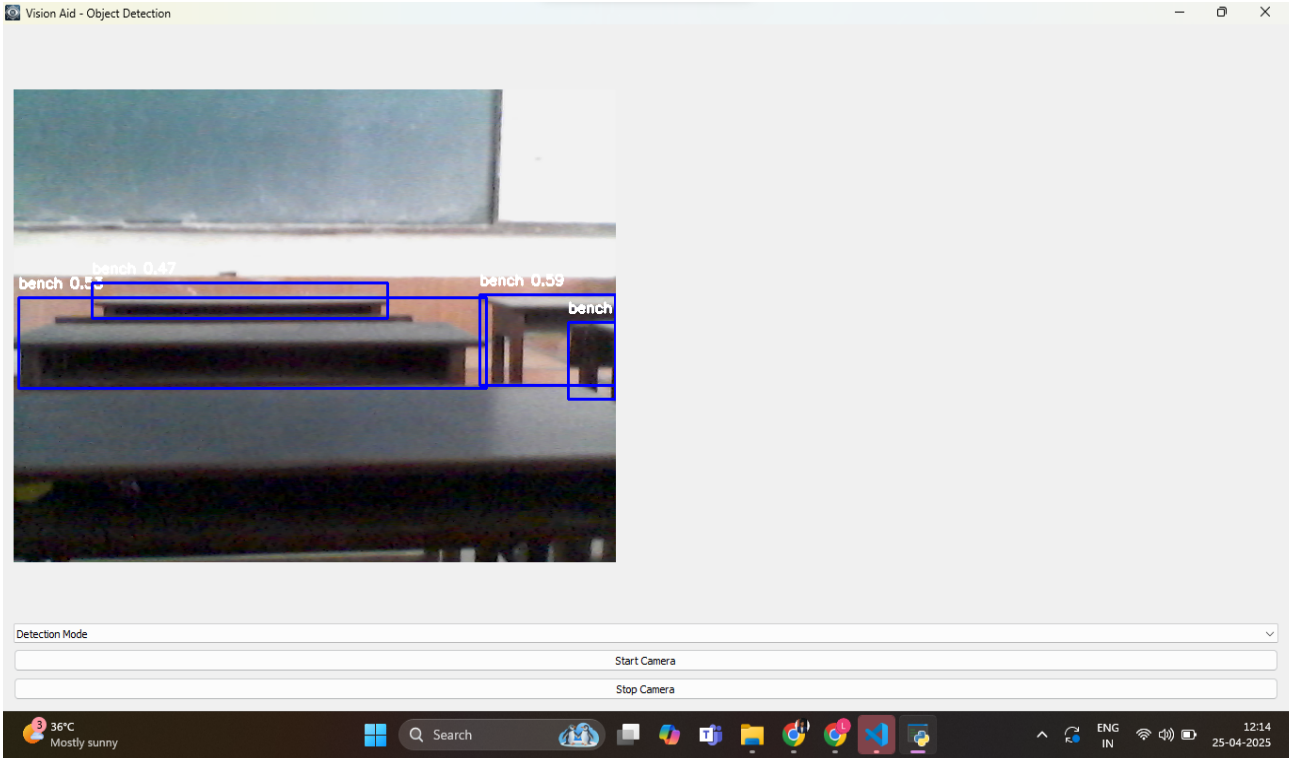

Figure 6.2: **Output with object recognition**

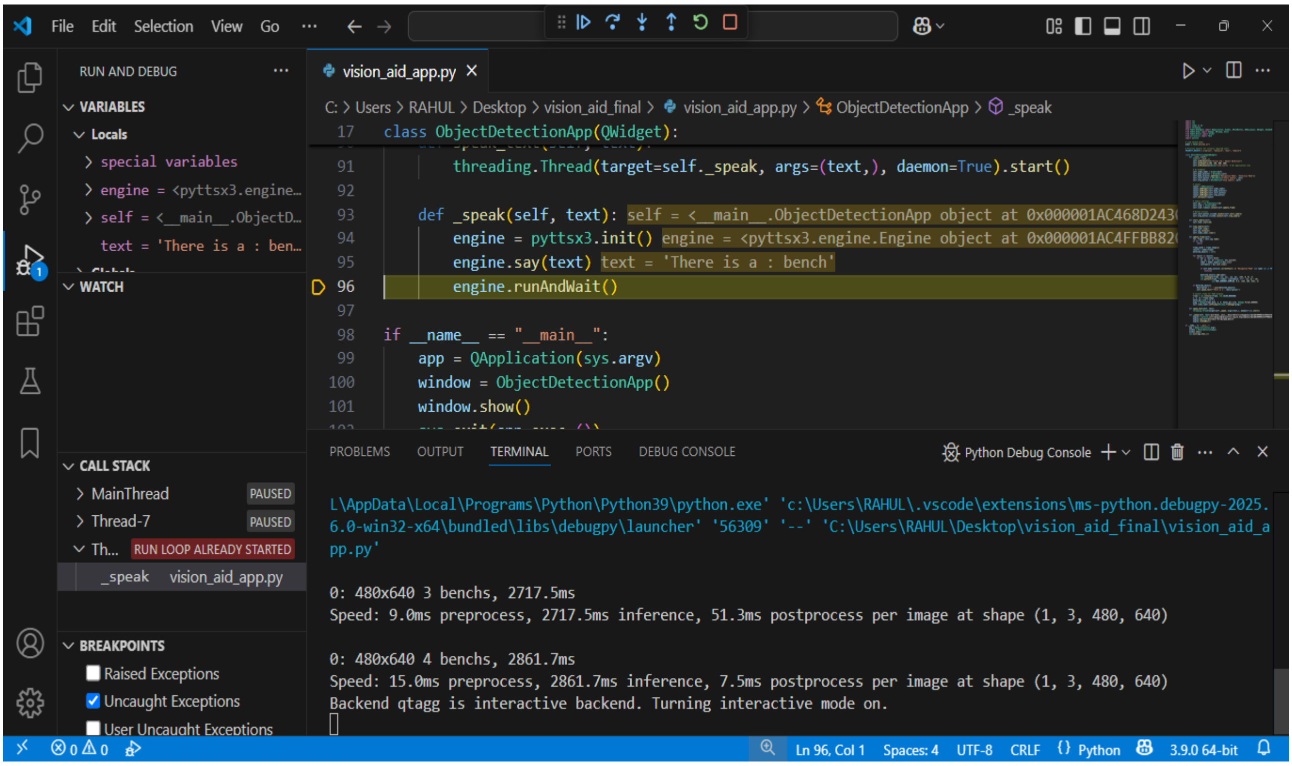

Figure 6.3: **Displays the recognized object in the output console.**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The project titled ”Real-Time Object Detection and Audio Guidance for Visually Impaired Individuals” represents a significant step forward in making technology more inclusive and supportive for people with visual impairments. By integrating computer vision with audio feedback mechanisms, this system provides an intelligent solution to assist users in navigating unfamiliar environments. The implementation of the YOLOv8 (You Only Look Once) model ensures highly efficient and accurate object detection in real-time. This is complemented by the text-to-speech module, which converts the detected object information into spoken words, offering an intuitive experience that requires no visual interface.

Furthermore, the user interface developed using PyQt5 enables users or caretakers to interact with the system through a graphical environment, providing options to start or stop detection, monitor system performance, and switch between modes. The modular architecture ensures scalability and allows developers to make modifications or add new features with ease. Overall, the system enhances autonomy, confidence, and safety for visually impaired individuals by providing real-time situational awareness. It successfully demonstrates how artificial intelligence and deep learning technologies can be leveraged to improve accessibility and daily living experiences for users who face visual challenges.

7.2 Future Enhancements

While the proposed system delivers reliable object detection and audio-based guidance, future versions could benefit from additional features that make the solution more dynamic, portable, and user-friendly. One potential enhancement involves integrating global positioning system (GPS) technology to offer context-aware navigation assistance. By incorporating location-based data, the system could direct users along safe walking paths, notify them of upcoming intersections, or even provide public transportation details. Another major improvement would be to miniaturize the hardware setup and embed it into wearable devices, such as smart glasses or pocket-sized cameras, reducing the need for the user to carry bulky equipment.

Moreover, future updates could implement adaptive machine learning algorithms that personalize object priorities based on user behavior or environment. For instance, the system might prioritize vehicle detection in outdoor settings or furniture in indoor spaces. The inclusion of distance estimation through stereo vision or ultrasonic sensors can also enhance the system's utility by warning users about nearby obstacles. Additionally, enabling cloud connectivity for storing data and receiving firmware updates can make the application more robust and easily maintainable. With multilingual text-to-speech support, emotional tone detection, and integration with smart home systems, the project can evolve into a complete lifestyle assistant for visually impaired individuals, improving both their mobility and quality of life.

Chapter 8

PLAGIARISM REPORT

Figure 8.1: **Plagiarism Report**

Appendices

Appendix A

Complete Data / Sample Data / Sample Source Code / etc

Appendix A

A.1 Sample Data

The application does not rely on pre-collected datasets for predictions in real-time but uses pre-trained YOLOv8 models for object detection. However, the sample input and output behavior for object detection is illustrated below.

Sample Input (Real-Time Frame from Webcam):

- Image Frame showing an indoor environment with a table, chair, and door.

Expected Output (From YOLOv8):

- Detected Objects: `[\chair", \table", \door"]`
- Audio Output: “Chair on your left”, “Table ahead”, “Door to your right”
- Haptic Feedback: Vibration intensity increases as user approaches an obstacle

YOLOv8 Model Details:

- Model: YOLOv8n (Nano version for fast inference)
- Framework: Ultralytics YOLOv8
- Classes: Common indoor objects (chair, table, door, wall, stairs)

A.2 Sample Audio Output

The system uses text-to-speech (TTS) to provide guidance. Below are a few example outputs:

- “Obstacle ahead”

- “Chair on the left”
- “Wall to your right”
- “Stairs ahead – proceed with caution”

A.3 Core Python Code Snippets

Object Detection (YOLOv8):

```
from ultralytics import YOLO
model = YOLO('yolov8n.pt')  # Load YOLOv8 model
results = model.predict(source='0')  # Real-time webcam detecti
for r in results:
    print(r.names, r.boxes)
```

Text-to-Speech Audio Feedback:

```
import pyttsx3
engine = pyttsx3.init()
engine.say("Chair on the left")
engine.runAndWait()
```

Simple PyQt GUI Initialization:

```
from PyQt5.QtWidgets import QApplication, QMainWindow
app = QApplication([])
window = QMainWindow()
window.setWindowTitle("Vision Aid")
window.show()
app.exec_()
```

A.5 Tools and Libraries Used

- **YOLOv8 (Ultralytics)** – Object detection model

- **OpenCV** – Real-time video frame capture
- **PyQt5** – GUI for desktop application
- **pyttsx3 / gTTS** – Text-to-speech
- **Python 3.10+** – Core programming language

A.6 Source Code

```python
import sys
import cv2
import numpy as np
import threading
from PyQt5.QtWidgets import QApplication, QLabel, QPushButton, QVBoxLayout, QWidget, QComboBox
from PyQt5.QtGui import QImage, QPixmap, QIcon
from PyQt5.QtCore import QTimer
from ultralytics import YOLO
import pyttsx3

# Load YOLOv8 model
model = YOLO("yolov8s.pt")

# Priority objects for visually impaired users
PRIORITY_OBJECTS = {"person", "obstacle", "door", "mobile"}

class ObjectDetectionApp(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("Vision Aid - Object Detection")
        self.setGeometry(100, 100, 800, 600)
        self.setWindowIcon(QIcon("icon.ico"))  # Set application icon

        # UI Elements
        self.video_label = QLabel(self)
        self.mode_selector = QComboBox(self)
        self.mode_selector.addItems(["Navigation Mode", "Detection Mode"])
        self.start_button = QPushButton("Start Camera", self)
        self.stop_button = QPushButton("Stop Camera", self)

        # Layout
        layout = QVBoxLayout()
        layout.addWidget(self.video_label)
        layout.addWidget(self.mode_selector)
        layout.addWidget(self.start_button)
        layout.addWidget(self.stop_button)
        self.setLayout(layout)

        # Camera settings
```

```
        self.cap = cv2.VideoCapture(0)
        self.timer = QTimer(self)
        self.timer.timeout.connect(self.update_frame)

        # Button actions
        self.start_button.clicked.connect(self.start_camera)
        self.stop_button.clicked.connect(self.stop_camera)

    def start_camera(self):
        self.timer.start(30)

    def stop_camera(self):
        self.timer.stop()
        self.cap.release()
        self.video_label.clear()

    def update_frame(self):
        ret, frame = self.cap.read()
        if not ret:
            return

        frame_width = frame.shape[1]
        results = model(frame)
        detected_objects = set()

        for result in results:
            for box in result.boxes:
                x1, y1, x2, y2 = map(int, box.xyxy[0])
                label = model.names[int(box.cls)]
                confidence = box.conf.item()

                if self.mode_selector.currentText() == "Navigation Mode" and label not in
                    PRIORITY_OBJECTS:
                    continue

                detected_objects.add(label)
                cv2.rectangle(frame, (x1, y1), (x2, y2), (255, 0, 0), 2)
                cv2.putText(frame, f"{label} {confidence:.2f}", (x1, y1 - 10),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)

        if detected_objects:
            description = ", ".join(detected_objects)
            self.speak_text(f"There is a : {description}")

        # Convert frame for PyQt display
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        h, w, ch = frame.shape
        bytes_per_line = ch * w
        qimg = QImage(frame.data, w, h, bytes_per_line, QImage.Format_RGB888)
        self.video_label.setPixmap(QPixmap.fromImage(qimg))
```

```python

    def speak_text(self, text):
        threading.Thread(target=self._speak, args=(text,), daemon=True).start()

    def _speak(self, text):
        engine = pyttsx3.init()
        engine.say(text)
        engine.runAndWait()

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = ObjectDetectionApp()
    window.show()
    sys.exit(app.exec_())
```

References

[1] W. Park, H. An and S. Park, ”Smart Docent System for Visually Impaired People with Mobile-Based Artificial Intelligence of Things,” in *TENCON 2024 - 2024 IEEE Region 10 Conference (TENCON)*, Singapore, 2024, pp. 1541-1544, doi: 10.1109/TENCON61640.2024.10902945.

[2] M. P, B. S. H. S, Y. A. P and A. N, ”Computer Vision Empowered Assistive Technology for Blind People,” in *2024 International Conference on Emerging Research in Computational Science (ICERCS)*, Coimbatore, India, 2024, pp. 1-7, doi: 10.1109/ICERCS63125.2024.10895124.

[3] E. Mohanraj, D. Devadarshini, M. Manonavaneethan and S. Aravind, ”Vision Assist: Real-Time Guidance System,” in *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*, Gobichettipalayam, India, 2024, pp. 714-721, doi: 10.1109/ICUIS64676.2024.10866444.

[4] D. Manju, B. Gollapalli, P. Sudheer Benarji, K. Pooja and A. Made, ”Performance Comparison of MobileNet SSD and YOLO v4 in Object Detection,” in *2024 IEEE 6th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, Hamburg, Germany, 2024, pp. 141-147, doi: 10.1109/ICCCMLA63077.2024.10871895.

[5] T. Acar, A. Solmaz, I. Cengiz and A. S. Bozkir, ”From Pixels to Paths: SIGHT - A Vision-Based Navigation Aid for the Visually Impaired,” in *2024 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Istanbul, Turkiye, 2024, pp. 1-8, doi: 10.1109/HORA61326.2024.10550694.

[6] A. Ajina, R. Lochan, M. Saha, R. B. K. Showghi and S. Harini, ”Vision Beyond Sight: An AI-Assisted Navigation System in Indoor Environments for the Visually Impaired,” in *2024 International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS)*, Bengaluru, India, 2024, pp. 1-6, doi: 10.1109/ICETCS61022.2024.10543550.

[7] B. Nataraj, K. R. Prabha, R. Abinaya, R. Divya Rani and V. S. Christina, ”Smart Cane with Object Recognition System,” in *2024 5th International Conference*

on Smart Electronics and Communication (ICOSEC), Trichy, India, 2024, pp. 1438-1443, doi: 10.1109/ICOSEC61587.2024.10722303.

[8] V. Bobby, S. Praneel, G. K. S, A. A. S and R. S. L, ”EBICS: Even Blind I Can See, A Computer Vision Based Guidance System For Visually Impaired,” in *2023 International Conference on Control, Communication and Computing (ICCC)*, Thiruvananthapuram, India, 2023, pp. 1-6, doi: 10.1109/ICCC57789.2023.10165192.

[9] K. K, Y. V. R. R, M. B. P. A. A, K. B. C. H and M. R, ”An Artificial Eye for Blind People,” in *2022 IEEE Delhi Section Conference (DELCON)*, New Delhi, India, 2022, pp. 1-5, doi: 10.1109/DELCON54057.2022.9752999.

[10] C. K. Lakde and P. S. Prasad, ”Navigation system for visually impaired people,” in *2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*, Melmaruvathur, India, 2015, pp. 0093-0098, doi: 10.1109/ICCPEIC.2015.7259447.