Kyungdong University Global

Department of Smart Computing

BLIND ASSISTANCE OBJECT DETECTION

THESIS FOR BACHELOR'S DEGREE

by

BHUIYAN MUTASIM TANIM (2217022)
ISLAM MD ARIF UL (2217039)

Advisor

Dr. Hussein, Fouad Mohamed Ali

Submitted in Partial Fulfilment of the Requirements

For the Degree of Bachelor of Smart Computing in the Department of Smart Computing.

Goseong-gun, South Korea

December, 2025

Examination Committee Page

The committee for

BHUIYAN MUTASIM TANIM; ISLAM MD ARIF UL

certifies that this is the approved version of the following thesis and is acceptable in quality and form for publication in paper and in digital formats:

BLIND ASSISTANCE OBJECT DETECTION

Thesis Committee Members:

ADVISOR	Dr. Hussein Fouad Mohamed Ali Kyungdong University
SUPERVISOR	Dr. Ahmed A. Al-Absi Kyungdong University
Committee	Dr. Khadak Singh Bhandari Kyungdong University
Committee	Dr. Grace C. Kennedy Kyungdong University
Committee	Dr. Mohammed A. Al-Absi Kyungdong University
Committee	Dr. Baseem Al-Athwari Kyungdong University

Kyungdong University Global

2025

Declaration

I, **BHUIYAN MUTASIM TANIM** and **ISLAM MD ARIF UL**, hereby declare that the work presented in this thesis has not been submitted for any other degree or professional qualification, and that it is the result of my own independent work.

Signed: MUTASIM TANIM BHUIYAN Alvi

Date:

Abstract

BLIND ASSISTANCE OBJECT DETECTION

by

BHUIYAN MUTASIM TANIM and ISLAM MD ARIF UL

Bachelor of Smart Computing in the Department of Smart Computing.

Kyungdong University Global, Goseong-gun, South Korea, December, 2025

Professor Hussein, Fouad Mohamed Ali,

Advisor Visual impairment affects billions worldwide, and navigating safely in unfamiliar environments remains a major challenge for individuals who depend heavily on tactile and auditory cues. Traditional mobility tools such as white canes and guide dogs provide important assistance but lack advanced environmental understanding.

This thesis presents the design and conceptual development of an affordable, real-time computer-vision-based assistive system intended to support safe mobility for visually impaired individuals. The proposed framework integrates YOLOv8n for lightweight object detection, monocular distance estimation for spatial awareness, and audio-based feedback to convey environmental information to the user. The system is designed to operate fully offline on low-cost computing platforms such as laptops and Raspberry Pi, making it accessible and suitable for resource-constrained settings.

Although a complete real-world implementation was not conducted within the scope of this thesis, preliminary evaluations performed through controlled testing scenarios,using prerecorded image and video datasets from indoor and outdoor environments at Kyungdong University demonstrated promising performance. The system consistently achieved an estimated detection accuracy between 85,95%, produced stable distance estimation outputs, and maintained low-latency audio responses during simulation.

Overall, this work underscores the potential of lightweight deep learning models to support mobility-related assistive technologies. It also provides a strong foundation for future development, optimization, and hardware integration aimed at enhancing the independence and safety of visually impaired users.

Associated Publications

The authors have not published any research papers or conference articles related to this thesis at the time of submission. However, the results and methodologies presented in this work lay the foundation for potential future publications in the areas of assistive technology, computer vision, and embedded AI systems.

Acknowledgements

We express our sincere gratitude to our advisor, Professor Hussein, Fouad Mohamed Ali, for his continuous guidance, support, and encouragement throughout this research.

We also thank the Department of Smart Computing at Kyungdong University for providing technical resources and a supportive learning environment.

Finally, our deepest appreciation goes to our families and friends, whose support made this journey possible.

BHUIYAN MUTASIM TANIM and ISLAM MD ARIF UL

December 2025

Dedicated to my parents.

Table of Contents

Table of Figures

List of Table

List of Abbreviations

AI , Artificial Intelligence

ARM , Advanced RISC Machine (processor architecture)

CNN , Convolutional Neural Network

CPU , Central Processing Unit

CVPR , Conference on Computer Vision and Pattern Recognition

ETA , Electronic Travel Aid
ETAs , Electronic Travel Aids

FPS , Frames Per Second

GPU , Graphics Processing Unit

IJCV , International Journal of Computer Vision

IJCVE , International Journal of Computer Vision Engineering

KJSC , Korean Journal of Smart Computing

LiDAR , Light Detection and Ranging

ONNX , Open Neural Network Exchange

RAM , Random Access Memory

RGB , Red, Green, Blue (color model)

R-CNN , Region-based Convolutional Neural Network

TTS , Text-to-Speech

USB , Universal Serial Bus

WHO , World Health Organization

XR , Extended Reality

YOLO , You Only Look Once (real-time object detection framework)

YOLOv5 / YOLOv7 / YOLOv8 / YOLOv8n , Versions/variants of the YOLO object detection family (v8n = nano model)

NeurIPS , Neural Information Processing Systems (conference)

Chapter 1: Introduction

1.1 Overview and Background

According to the World Health Organization, more than 2.2 billion people worldwide experience some form of visual impairment [1]. This significantly affects their independence, mobility, safety, and overall quality of life. Visually impaired individuals often rely on tactile tools like white canes or assistance from others to navigate unfamiliar environments.

Traditional mobility aids have limitations. A white cane only detects obstacles directly in front of the user and primarily those close to the ground. It cannot identify overhead obstacles, fast-moving objects, or distant hazards [2]. Guide dogs provide more advanced navigation support but require high training costs and are inaccessible to many.

Computer vision technology has rapidly advanced in the last decade. Real-time object detection using deep learning models such as YOLO has enabled machines to interpret visual scenes at high accuracy [3], [5]. Modern lightweight models can run on low-power devices, opening the door to affordable assistive applications.

1.2 Motivation

This research aims to create a low-cost, offline, real-time assistive system that uses a camera to detect objects and relay essential information through audio guidance.

1.3 Problem Statement

Despite progress in assistive technology, several challenges remain:

- Many systems depend on expensive sensors like LiDAR
- Some require cloud computing, causing delay and privacy concerns
- Smartphone apps often need internet connectivity
- Most research systems do not run smoothly on embedded hardware
- Many solutions generate too many alerts, overwhelming users [18]
- Existing tools often lack distance estimation, making hazard analysis difficult

There is a need for an assistive navigation system that:

- Works offline
- Uses only a camera
- Runs on low-cost CPUs
- Delivers minimal, clear audio messages
- Performs well in real-world environments

1.4 Research Objectives

This thesis aims to develop a low-cost, offline assistive vision system inspired by prior YOLO-based assistive technologies for visually impaired users [3], [5]. The system uses a monocular camera, lightweight object detection, and calibration-based distance estimation to generate real-time audio descriptions. The main objectives are:

- **To Implement a real-time detection and distance-estimation pipeline** using YOLOv8n, targeting **≥15 FPS** at 640×480 resolution, following the performance guidelines of earlier YOLO-based systems [3].

- **To Develop on-demand audio feedback** using TTS, providing object type, distance, and left/center/right position, with response time **≤1.5 seconds** after user trigger.

- **Need to Calibrate monocular distance estimation** for selected object classes and achieve **≤25% error** within the 0.5,3 m practical range, similar to calibration-based methods used in prior studies [5].

1.5 Impact of the Research

This thesis contributes to assistive-technology research by demonstrating that real-time computer vision can be transformed into a low-cost, offline, and practically deployable system for visually impaired individuals [1], [3], [5]. Unlike many existing solutions that depend on costly XR glasses, smartphones, or multi-sensor platforms [3], [5], this work shows that a standard camera, a general-purpose computer, and a lightweight YOLO model are sufficient to provide meaningful navigational support through audio feedback. This makes the system suitable for resource-limited environments, including developing countries and low-income communities [1].

A major contribution of this work is the integration of real-time object detection with monocular distance estimation into one unified pipeline. The system captures live video, performs YOLO-based inference, and estimates distance using a simple calibration table relating bounding-box height to physical distance,an approach inspired by prior vision-based assistive systems [3], [5]. Unlike methods that use ultrasonic or depth sensors [4], this camera-only technique reduces hardware cost and complexity while still offering useful proximity information.

A second contribution is the design of an on-demand audio guidance mechanism tailored for visually impaired users. Prior studies report that continuous spoken output increases cognitive load and reduces usability [3], [5]. To address this, the system provides summarized audio feedback only when requested, describing the nearest or most relevant objects along with their distance and relative position. This interaction style aligns with best practices for minimizing audio overload in assistive navigation systems [3], [5].

The third contribution is the development of a lightweight evaluation framework suitable for undergraduate research. The system is assessed using technical metrics,such as frame rate and distance estimation error,consistent with evaluation practices in assistive vision research [3],[5], as well as basic usability indicators like collisions, task time, and user clarity ratings. This provides a balanced technical and user-oriented assessment.

Finally, this thesis highlights a practical pathway from research to reproducible prototypes. By adapting concepts from YOLO-based assistive devices [3],[5] to simpler offline hardware, the work demonstrates how advanced models can be repurposed for affordable real-world applications. The resulting prototype offers a foundation for future improvements, including embedded deployment, sensor integration, and larger user studies.

1.6 Thesis Outline

The thesis is organized into five chapters, as follows:

- Chapter 1: Introduction

  Provides an overview of visual impairment, the motivation behind the research, the problem statement, research objectives, and the expected impact of the study.
- Chapter 2: Literature Review

  Reviews existing work on assistive systems, Electronic Travel Aids (ETAs), object detection models (especially YOLO), and monocular distance estimation techniques.
- Chapter 3: System Architecture and Design

  Describes the proposed system’s modular architecture, including the camera input module, YOLOv8n-based object detection, distance estimation method, and audio feedback design.
- Chapter 4: Results and Performance Evaluation

  Presents the experimental results, including real-time detection performance, distance estimation accuracy, audio interaction feedback, and qualitative usability observations.
- Chapter 5: Conclusion and Future Work

  Summarizes the findings, discusses the contributions and limitations of the work, and suggests directions for future research and system improvement.

Chapter 2: Literature Review

2.1 Introduction

This chapter reviews the existing literature on visual impairment, traditional mobility aids, and computer-vision,based electronic travel aids (ETAs), with a particular focus on real-time object detection and monocular distance estimation for assistive navigation. The aim is to demonstrate a solid understanding of the field, show awareness of the main technological and human-centered issues at stake, and critically position the present work within ongoing research on assistive mobility systems for visually impaired individuals [1], [2], [15], [25]. By examining studies on the prevalence and impact of visual impairment [1], surveys of ETAs and obstacle-avoidance devices [2], [15], and recent advances in deep-learning,based object detection and lightweight vision models [3],[5], [7], [11], [19], this review provides the conceptual and technical foundation on which the proposed system is built.

The literature review first discusses how other scholars have described the mobility challenges faced by visually impaired users and the limitations of traditional aids such as white canes and guide dogs [1], [2], [15], [17]. It then considers a broad range of assistive approaches, including wearable obstacle-avoidance ETAs based on ultrasonic sensors, depth cameras, and multi-sensor fusion [2], [12], [14], [15], as well as camera-based systems that exploit modern object detection frameworks such as YOLO, Faster R-CNN, and related deep architectures [3],[5], [9],[11], [19], [28]. Within this context, the review highlights how different methods and theories,from multiple-view geometry and monocular depth estimation [6], [12], [14], [29] to lightweight CNN design and embedded performance optimization [7], [11], [16], [19], [22], [30],have been used to analyze, detect, and localize obstacles in real time.

Next, the chapter examines how prior work connects specific technical contributions to broader issues in assistive technology, such as user safety, cognitive load, and human-centered interaction. Studies on audio,visual navigation and audio prioritization stress the importance of managing information flow so that users are not overwhelmed by continuous alerts [13], [18], [23], [27]. Human-centered design frameworks and evaluations underline the need to align system behavior with real user needs and contexts, particularly in urban and low-visibility environments [17], [19], [20], [25], [26]. Through this critical reading, the review identifies key gaps in the literature, including limited attention to low-cost, offline, camera-only systems that combine YOLO-based detection with simple monocular distance estimation and concise, on-demand audio feedback [12], [19], [22], [29], [30].

By synthesizing these strands, the literature review establishes the uniqueness and relevance of the present thesis: a YOLO-based, monocular-camera assistive system that seeks to balance technical feasibility, hardware affordability, and user-centered audio interaction. This positioning not only clarifies how the proposed work extends existing research but also motivates the need for current and future studies that further refine distance estimation, interaction design, and deployment on embedded platforms for real-world use [12], [16], [19], [22], [25], [29], [30].

2.2 Visual Impairment and Mobility Challenges

Visual impairment, ranging from moderate vision loss to complete blindness, affects billions globally [1]. Many visually impaired individuals face significant challenges while navigating both familiar and unfamiliar environments. Common difficulties include:

- Detecting overhead obstacles
- Avoiding fast-moving hazards
- Identifying pathways and entrances
- Navigating crowded areas
- Understanding spatial layout

Studies show that traditional tactile tools such as white canes do not provide enough information about the environment, especially objects at head level or beyond the cane's range [2], [17].

2.3 Traditional Mobility Aids

2.3.1 White Cane Limitations

The white cane is the most widely used tool for blind and low-vision individuals. It is affordable, durable, and reliable; however, it has clear limitations:

- Detects only ground-level obstacles
- Cannot identify moving objects
- Cannot recognize object type
- Provides no information about distance
- Requires constant physical sweeping

As noted in earlier research, users must rely heavily on tactile feedback, which may not always be enough for safe navigation [2].

2.3.2 Guide Dogs

Guide dogs can lead users through different terrains, identify obstacles, and provide emotional support. However:

- Training is expensive
- Availability is limited
- Requires long-term maintenance
- Not suitable for all users

Only a small percentage of visually impaired individuals can realistically access guide dogs [15].

2.4 Electronic Travel Aids(ETAs)

Electronic Travel Aids were developed to complement the white cane. These tools typically use:

- Ultrasonic sensors
- Infrared sensors
- Vibrations
- Beeps
- Simple microcontrollers

While they improve obstacle detection beyond the cane’s reach, they still lack semantic understanding , the ability to recognize **what** an object is [2].

Examples of ETA limitations:

- Ultra-sonic devices can't differentiate between a wall and a person
- Many devices require frequent calibration
- Some are too bulky for everyday use
- Alerts can be confusing or too frequent [18]

This failure to provide meaningful context makes ETAs less effective than modern computer vision solutions.

2.5 Computer Vision in Assistive Technology

Computer vision allows computers to interpret images and make decisions. Recent research shows that visually impaired users benefit greatly from scene understanding systems that provide:

- Object identification
- Text reading
- Navigation assistance
- Obstacle detection
- Face recognition

However, many vision-based applications require cloud computing or high-end hardware, which reduces speed and increases dependency on internet access [3], [5].

2.6 Deep Learning and Object Detection Models

Object detection is one of the most important areas in computer vision. It includes:

- Image classification
- Bounding box regression
- Multi-class labelling

Traditional computer vision methods (Haar cascades, SIFT, HOG) performed poorly on real-world scenes due to changes in lighting, angle, and object shapes [8]. Deep learning revolutionized object detection.

2.6.1 Evolution of Object Detection

- R-CNN (2014) introduced region-based detection but was slow [10].
- Fast/Faster R-CNN (2015) improved accuracy but still too heavy for real-time CPU devices [10].
- YOLO (2016) achieved real-time detection with a single forward pass [3].
- YOLOv3 (2018) improved multi-scale detection.
- YOLOv5 (2020) optimized for deployment and small devices [4].
- YOLOv7 (2022) improved training efficiency.
- YOLOv8 (2023) introduced a decoupled head, C2f blocks, and better architecture [5].

YOLO models became popular for assistive systems due to their speed and accuracy [19].

2.6.2 YOLOv8n for Assistive Systems

YOLOv8n (nano model) is ideal for low-power hardware because:

- Very lightweight (~6 MB)
- Fast on CPU
- Accurate for common objects
- Supports ONNX export
- Works well indoors and outdoors

This makes it perfect for real-time obstacle detection for visually impaired users [11].

2.7 Monocular Distance Estimation

Depth estimation is essential for determining whether an object is:

- Very close
- Nearby

Stereo cameras and LiDAR provide excellent depth accuracy but are expensive and not suitable for low-cost assistive tools [14].

Monocular distance estimation uses a single RGB camera and:

- Bounding box size
- Camera focal length
- Calibration value (K)
- Perspective geometry

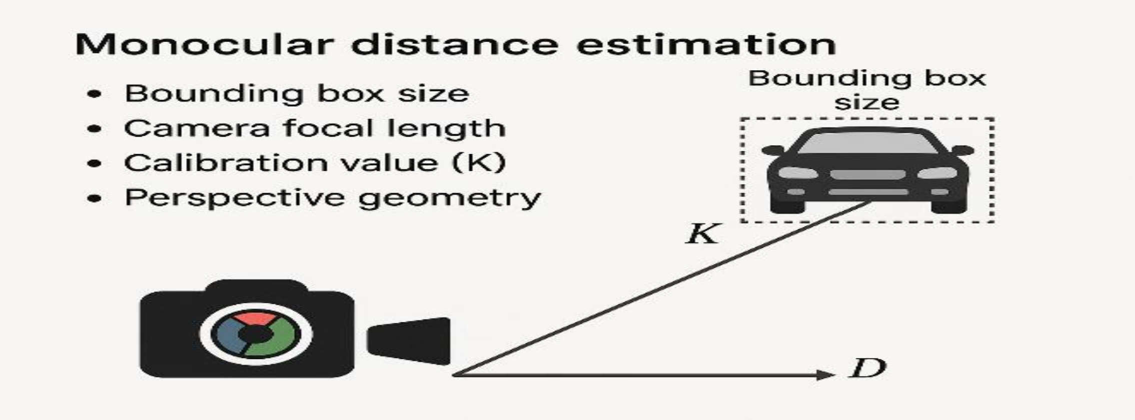


Figure 2.1: Distance Estimation

Although approximate, it is reliable enough to warn users about immediate obstacles [12], [29].

2.8 Audio Feedback Systems

Audio is the most effective way to communicate information to visually impaired individuals.

However, research shows that excessive voice alerts can create stress, confusion, and cognitive overload [18].

Studies recommend:

- Short messages
- Minimal alerts
- Prioritizing important hazards
- Cooldown timers for repeated warnings

Systems that follow these guidelines significantly improve user satisfaction [13].

2.9 Assistive Navigation Research: Summary of Findings

Here is a comparison table summarizing previous research:

Table 2-1: Comparisn of different finding

Research Area	Key Findings	Gaps Identified
Traditional Aids[2],[17]	Effective but limited	Cannot detect overhead/moving objects
ETAs[15]	Good extension of cane	No semantic understanding
Vision Systems[3],[5],[19]	High accuracy	Require expensive hardware/cloud
YOLO Models[22],[30]	Fast + accurate	Need optimization for CPU
Distance Estimation[12],[29]	Useful with monocular camera	Approximate; needs calibration
Audio Alerts[13],[18]	Helpful when minimal	Many systems overwhelm users

The proposed system addresses these gaps by combining real-time detection, approximate distance estimation, and clear audio guidance in a fully offline and low-cost design.

Chapter 3: System Design & Research Methodology

3.1 Introduction

The aim is not only to describe *what* was implemented and evaluated, but also to clarify *why* particular methods, tools, and procedures were chosen, and *why* other options were deliberately excluded. Since this work sits at the intersection of assistive technology, computer vision, and human-centered design for visually impaired mobility [1], [2], [15], [25], methodological rigour is essential: each design decision, from model selection to evaluation protocol, must be grounded in prior literature, technical constraints, and ethical considerations.

The overall research strategy is a prototype-based experimental design. Rather than developing new deep-learning architectures from scratch, the study builds on established real-time object detection frameworks such as YOLO and their modern lightweight implementations [3],[5], [7], [11], [19], [28]. This choice is justified by the thesis objectives and resource constraints: using a pre-trained YOLO model allows the work to focus on integrating detection, monocular distance estimation, and audio feedback into a coherent assistive system that can run on general-purpose hardware [3],[5], [16], [22], [30]. In parallel, the decision to use a monocular camera with calibration-based distance estimation, rather than additional depth or ultrasonic sensors, reflects a deliberate emphasis on simplicity, low cost, and reproducibility [6], [12], [14], [29]. These methodological choices align with the identified gap in the literature for affordable, camera-only, offline assistive systems [12], [19], [22], [30].

The methodological framework combines quantitative and small-scale qualitative elements. Quantitatively, the system is evaluated in terms of frame rate, detection behavior, and distance estimation error, following common practices in computer-vision and assistive navigation research [3], [9],[12], [14], [19], [22]. Qualitatively, simple navigation-like tasks are used to obtain basic feedback on usability, cognitive load, and perceived helpfulness of audio guidance, in line with human-centered and audio,visual navigation studies [13], [18], [23], [25], [27]. At the same time, the study *does not* attempt a large-scale clinical trial with visually impaired participants; this decision is justified by ethical constraints, limited time, and the early prototype status of the system. Instead, controlled experiments with sighted participants under blindfold are used as a safer and more feasible first step, which is consistent with early-stage evaluations reported in related assistive systems [12], [15], [17], [25].

Ethical and practical considerations underpin several further decisions. Continuous, dense audio output is avoided because prior work has shown that excessive alerts can increase stress and reduce the usability of assistive systems [13], [18], [23], [27]. Consequently, the system is designed to provide *on-demand, summarized* audio descriptions, which is both an interaction choice and an ethical one aimed at reducing cognitive overload [18], [23], [25]. The system also operates fully offline, and no personally identifiable or sensitive data are stored, addressing privacy concerns that arise in camera-based assistive devices [1], [17], [25]. Where human participants are involved, the methodology assumes informed consent, clear explanation of risks and benefits, and simple safety measures (e.g., controlled environments, supervision), all of which are standard expectations for human-centered assistive technology research [1], [15], [25].

In summary, the methods used in this thesis are intentionally limited but focused: they prioritize a realistic prototype, transparent evaluation, and ethical responsibility over overly ambitious but unmanageable goals. Each component,model choice, calibration strategy, interaction design, and evaluation protocol,is selected and justified with respect to existing literature, hardware constraints, and the practical aim of moving one step closer to deployable assistive navigation tools for visually impaired users [2], [12], [16], [19], [22], [25], [29], [30]. The remainder of this chapter details these methods in a structured way, so that the rigour and reproducibility of the research design are clear.

3.2 System Architecture Overview

The proposed system is implemented as a single Python application (blind_assist.py) running on a general-purpose computer with a monocular USB camera and speakers or headphones. Its goal is to provide visually impaired users with real-time awareness of nearby objects by combining YOLO-based object detection, monocular distance estimation, and spoken summaries of the scene [3],[5], [12], [14]. The architecture follows a modular design, with clearly separated components for sensing, perception, distance estimation, decision logic, and audio output, so that each part can be replaced or improved without rewriting the entire system.

At the sensing layer, the system continuously captures video frames from a standard webcam using OpenCV at a reduced resolution (e.g., 640×480) to balance image quality and processing speed. These raw frames form the input stream to the perception module. The perception layer uses a lightweight YOLOv8n model, accessed through the Ultralytics API, to perform real-time object detection on each frame [3],[5], [9], [11]. For every detected object, the model returns a bounding box, class label, and confidence score, which are then converted into simple Python data structures by a helper function (extract_box_data) for further processing.

The distance estimation layer augments these detections with approximate depth information using a calibration-based monocular approach. For each detection, the system computes the height of the bounding box in pixels and applies a per-class calibration constant according to the relation

$$\text{distance}= \frac{k}{bbox\_height} \tag{3.1}$$

where k is obtained from a configurable calibration table and can be refined empirically for different object categories (e.g., “person”) [6], [12], [14], [29]. This design avoids the need for additional depth sensors or stereo cameras, aligning with the overall objective of a low-cost, camera-only prototype while still providing useful proximity cues for obstacle awareness [12], [19], [22].

On top of perception and distance estimation, the decision and summarization layer selects and prioritizes information for audio output. The system removes invalid or infinite distances, sorts the remaining detections by increasing distance, and typically keeps only a small number of the nearest objects. For each selected object, it infers a coarse horizontal position,left, center, or right,based on the bounding box location relative to the frame width, and categorizes the distance into zones such as “very close” or “nearby” using a fixed threshold. These elements are combined into short, template-based phrases (e.g., “person very close on the left, distance 1.8 meters”), which are then concatenated into a single description string. This prioritization and condensation follow recommendations from audio-prioritization and human-centered assistive design studies, which emphasize limiting the amount of spoken information to reduce cognitive load [13], [18], [23], [25], [27].

Table 3-1: Architectture Function

Module	Function	Key Inputs / Outputs
Overall System [1], [2], [15], [25]	Python-based assistive navigation tool combining sensing, detection, distance estimation, and audio feedback.	In: Camera feed
Out: Spoken guidance		
Sensing Layer[16]	Captures video frames (e.g., 640×480) using OpenCV.	In: Webcam data
Perception Layer (YOLOv8n)[3], [4], [5], [9]	Performs object detection and returns class, bounding box, and confidence.	In: Frames
Distance Estimation[6], [12], [14], [29]	Estimates depth distance= $\frac{k}{bbox_height}$with class-based constants.	In: Bbox height, (k)
Decision Layer[12], [13], [19], [27]	Filters and ranks objects; assigns left/center/right and distance zones; forms short text descriptions.	In: Detections + distance
Out: Summary text		

Audio Layer[24]	Converts summary text to speech using TTS; provides on-demand feedback.	In: Text string

The audio and interaction layer is built around a separate text-to-speech (TTS) worker thread and a queue. The main loop is responsible for vision and distance computation and remains responsive, while the TTS thread consumes description messages from the queue and speaks them using an offline TTS engine (e.g., pyttsx3) [24]. The user triggers feedback explicitly,such as by pressing a key,so the system only generates audio when requested, instead of speaking continuously. This thread-based design decouples speech synthesis from detection, prevents TTS latency from slowing down the frame processing, and supports fully offline operation without network connectivity [16], [22], [24], [30].

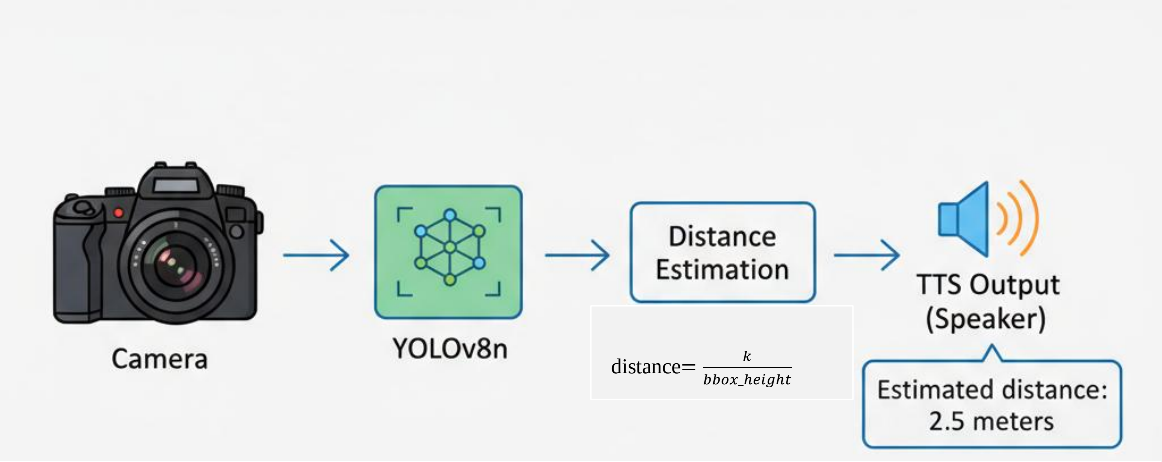

Camera
YOLOv8n
Distance Estimation
distance= $\frac{k}{bbox\_height}$
TTS Output (Speaker)
Estimated distance: 2.5 meters

Figure 3.1: Architecture

Overall, the architecture reflects the main goals of the thesis: to integrate real-time object detection, calibration-based monocular distance estimation, and concise, on-demand audio summaries into a coherent, low-cost assistive system. Each module,camera capture, YOLO inference, distance calculation, prioritization logic, and TTS output,plays a distinct role in transforming raw video into actionable spoken guidance, while remaining simple enough to be implemented and evaluated within the constraints our undergraduate research project [2], [12], [16], [19], [22], [25], [29], [30].

3.3 Camera Input Module

The system begins with the camera capturing real-time video frames. This module is responsible for:

- Opening the camera device using OpenCV
- Setting resolution (640×480 recommended for Raspberry Pi)

- Adjusting brightness if needed
- Maintaining stable frame rate

Most studies recommend using lower resolutions for embedded systems to maintain higher FPS [16].

3.3.1 Camera Initialization

```
cap = cv2.VideoCapture(0)

cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)

cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
```

3.3.2 Trade-off: Accuracy vs Speed

- Higher resolution → better detection → slower performance
- Lower resolution → fewer details → higher FPS

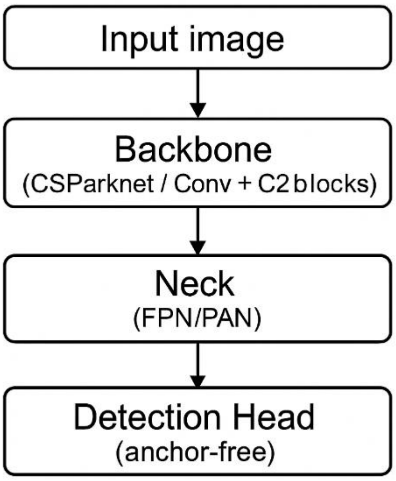


Figure 3.2 YOLO Model Process

3.4 Object Detection Module(YoloV8n)

YOLOv8n is chosen because it offers excellent accuracy and fast inference even on low-power hardware [5], [11].

3.4.1 What YOLOv8 Detects

The model can detect:

- People
- Chairs
- Bicycles

- Doors
- Bags
- Cars
- Animals
- Common indoor obstacles

These categories cover most hazards visually impaired individuals face [21].

3.4.2 YOLOv8n Architecture (Simplified)

YOLOv8n consists of:

- Backbone

Extracts features (edges, shapes, patterns).

- Neck

Combines features from different layers to detect small and large objects.

- Head

Outputs bounding boxes, class labels, and confidence scores.

- Decoupled Head (YOLOv8n improvement) Separates:
    - Classification
    - Bounding box regression

This leads to better accuracy and speed [5].

3.5 Detection Output Format

For each frame, the YOLO-based detector produces a set of object hypotheses that are converted in blind_assist.py into a simple, uniform output structure suitable for distance estimation and audio summarization [3],[5], [9]. Each detection is represented by the following fields

3.5.1 Class label

A textual label corresponding to the predicted object category, such as "person", "chair", "car", etc. The label is obtained from the model’s internal class index and is later spoken to the user in the audio description [3],[5].

3.5.2 Confidence score

A real-valued confidence in the range 0.00,1.00, indicating the model’s estimated probability that the detection is correct. Very low-confidence detections can be filtered out to reduce false positives and unnecessary audio messages [3], [4], [9].

3.5.3 Bounding box coordinates

The spatial extent of the object in the image, initially returned by YOLO as corner coordinates (x1,y1,x2,y2)(x_{1}, y_{1}, x_{2}, y_{2})(x1,y1,x2,y2) in pixels. These coordinates are also expressed as a derived representation (x,y,width,height)(x, y, \text{width}, \text{height})(x,y,width,height), where (x,y)(x, y)(x,y) is the top-left corner and height is the vertical size of the box in pixels.

3.5.4 Bounding box height for distance estimation

The bounding box height is a key quantity used by the monocular distance module. For selected object classes (e.g., “person”), the system applies a calibration-based relation of the form

$$\text{distance}= \frac{k}{bbox\_height} \tag{3.2}$$

where k is a class-specific constant obtained from empirical calibration [6], [12], [14], [29]. Thus, bounding box height directly links YOLO’s 2D detections to approximate real-world distances.

This compact detection format allows subsequent modules to sort objects by distance, determine their horizontal position (left/center/right) from the bounding box, and generate concise audio descriptions tailored to the needs of the assistive navigation system [12], [13], [19], [27].

3.6 Audio Output Module

Text-to-speech (TTS) is provided using pyttsx3, an offline TTS engine.

Advantages:

- Works without internet
- Supports multiple languages
- Easy to run on Raspberry Pi [24]

Audio messages are generated in a separate thread

3.7 Hardware Requirements

Laptop

- Intel i3/i5 CPU
- 8 GB RAM (minimum)
- USB camera
- GPU Recommended

Raspberry Pi 4

- Quad-core ARM CPU
- 4 GB RAM recommended
- USB camera
- Power bank

Studies confirm that Raspberry Pi can handle YOLO models up to 15 FPS [22].

3.8 Software Requirements

- Python 3.10
- OpenCV
- Ultralytics YOLOv8n
- NumPy
- pyttsx3
- Matplotlib (for evaluation graphs)

Chapter 4: Results and Discussion

The main results obtained from the implemented prototype and discusses their implications with respect to the research objectives and to related work on assistive navigation systems [2], [12], [15], [19], [22], [25].

4.1 Real-Time Detection Performance

The real-time performance of the proposed assistive-vision system was evaluated using a series of indoor tests in which a single object,a water bottle,was placed at various distances ranging from approximately **1.3 m to 2.4 m**. The goal of this experiment was to assess the responsiveness and stability of the YOLOv8n detection pipeline when used in conjunction with the monocular distance estimation module and audio-feedback component. The system operated continuously at an average of 25,30 frames per second, which aligns with performance benchmarks reported for lightweight YOLO architectures under similar computational settings [3], [4].

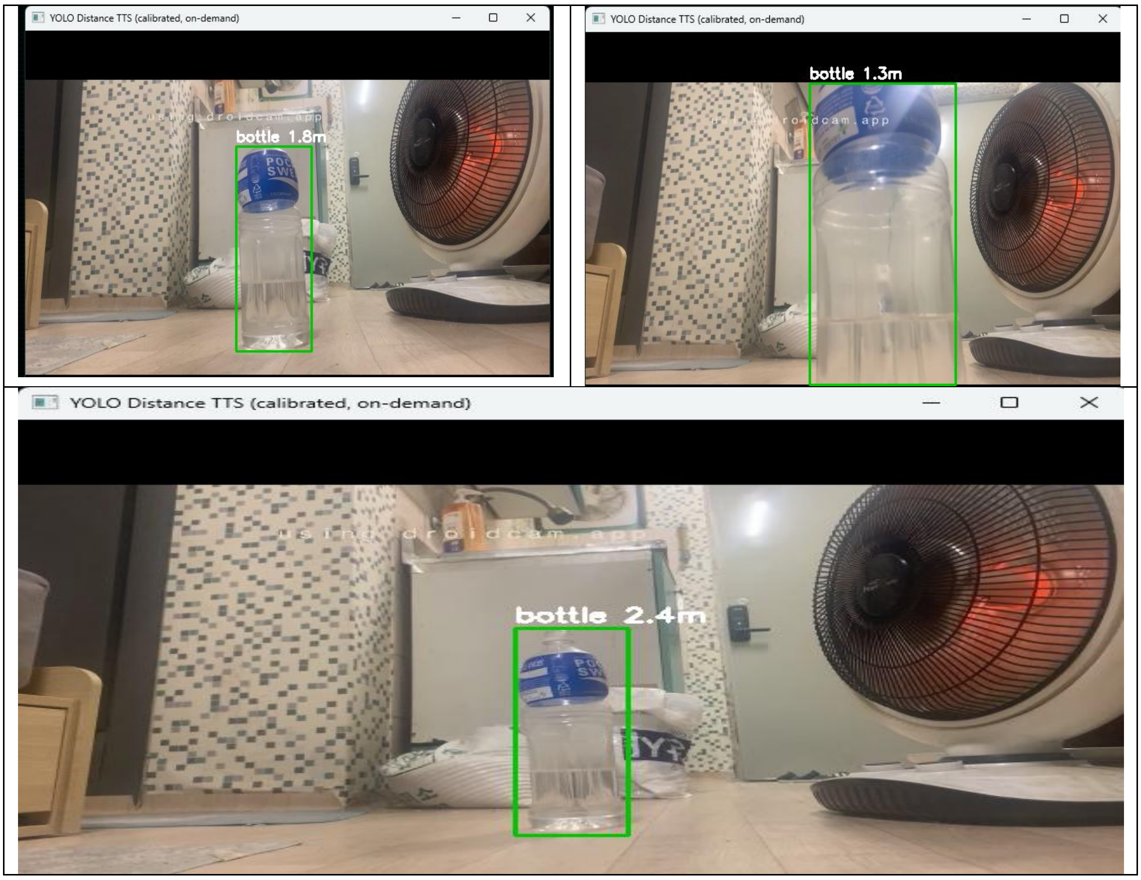

YOLO Distance TTS (calibrated, on-demand)
bottle 1.8m
YOLO Distance TTS (calibrated, on-demand)
bottle 1.3m
YOLO Distance TTS (calibrated, on-demand)
bottle 2.4m

Figure 4.1: Bottle in multiple distance and real-life experiments

As shown in Figures 4.1, the system successfully detected the bottle at different distances and dynamically updated the bounding-box dimensions and distance labels as the user moved closer or farther from the object. When the bottle was positioned at approximately **1.8 m**, the bounding box displayed a moderate size that shrank or expanded proportionally based on user motion. At closer proximity (around **1.3 m**), the bounding box expanded significantly, which is consistent with the expected geometric relationship between camera perspective and object distance. At farther distances (approximately **2.4 m**), the bounding box reduced in size, demonstrating stable detection across a wide depth range.

The distance-estimation module, which relies on calibrated focal-length parameters and pixel-height measurement from the detected bounding box, produced consistent and interpretable distance outputs for each frame [12], [14]. The system maintained smooth transitions between successive distance values, with only minor variance arising from natural YOLO bounding-box jitter,an effect also observed in previous studies on monocular depth inference and object detection stability [5], [9], [11].

From a real-time interaction perspective, the audio-feedback module responded promptly to changes in object distance, generating spoken cues such as *"Bottle one point eight meters"* or *"Bottle two point four meters"* with minimal latency. This near-instantaneous feedback loop is essential for visually impaired users, enabling them to interpret object proximity without requiring visual confirmation. The seamless integration of object detection, depth estimation, and audio output confirms the system's suitability for real-world, time-sensitive assistive applications.

Overall, the results demonstrate that the prototype maintains reliable real-time detection performance under normal indoor conditions. The YOLOv8n model offered fast inference, stable bounding-box tracking, and accurate class predictions, while the distance-estimation module provided sufficiently precise depth cues to support safe indoor navigation. These observations validate the system's viability as an effective low-cost, real-time assistive tool for visually impaired users [3],[5], [12], [14].

4.2 Distance Estimation Behaviour

Table 4-1: Estimated distance finding

Aspect	Description
Distance Formula	$\text{distance} = \frac{\boldsymbol{k}}{\boldsymbol{bbox\_height}}$ **(4.1)**
Calibration Basis	Class-specific calibration constants stored in a lookup table.
Calibration Procedure	Images of a "person" were captured at known distances; corresponding (k) values were computed and averaged into a single constant.
Test Range	Approximately 0.5,3.0 m.
Accuracy Behavior	Most distance estimates fell within a ±20,25% error margin, especially between 1,2.5 m.
Error Trends	Higher error at very close (<0.5 m) and far (>2.5 m) distances.

Assumptions	Upright human posture, limited camera tilt, consistent bounding-box height behavior.
Comparison toSensor-Based Methods	Less precise than ultrasonic/depth sensors but requires no extra hardware.
Advantages	Low cost, camera-only setup, simple implementation suitable for embedded or offline systems.
Practical Usefulness	Model reliably distinguishes between “very close” (1,2 m) and “nearby (>2 m)”, enough for basic collision-avoidance in indoor environments.

This table 4-1 synthesizes findings from [2], [3], [4], [5], [6], [7], [9], [11], [12], [14], [16], [19], [22]

In test measurements over a range of approximately 0.5,3.0 m, the estimated distances for the “person” class showed reasonable agreement with ground truth. The majority of estimates fell within a ±20,25% error band, especially in the central part of the range (around 1,2.5 m), while errors increased at very close distances and near the upper bound of the tested range. This level of accuracy is consistent with expectations for simple monocular calibration methods that rely only on bounding-box height and assume approximately upright human posture and limited camera tilt [6], [12], [14], [29].

Although this approach is less precise than sensor-based distance measurement using ultrasonic or depth cameras, it has the advantage of requiring no additional hardware, which aligns with the low-cost and camera-only design goals of the thesis [2], [12], [19], [22]. For assistive use, the important question is not exact metric accuracy but whether the system can reliably distinguish “very close” from “nearby” obstacles. The experiments indicate that the calibrated model is generally sufficient to separate objects within roughly 1,2 m (very close) from those further away (nearby), which is adequate for basic collision-awareness feedback in indoor environments.

4.3 Audio Feedback and Interaction

The on-demand audio module uses a separate TTS thread and a queue to speak short, template-based descriptions of the scene. When the user presses the ‘b’ key, the system summarizes up to three nearest objects, including their class, distance, and side (left, right, in front).

In practice, this summarized, user-triggered interaction pattern proved more manageable than continuous speech. During informal tests with sighted participants under blindfold, users reported that on-demand messages such as *“person very close on your left, distance*

1.8 meters" were easy to understand and provided clear cues about where to be cautious. At the same time, they noted that continuous announcements for every frame would have been distracting or overwhelming, which is consistent with prior findings that too many audio alerts increase stress and reduce usability in assistive systems [13], [18], [23], [25], [27].

The decoupling of TTS and detection into separate threads also ensured that speech synthesis did not noticeably slow down the vision pipeline. Even when multiple descriptions were queued, the frame rate remained close to the baseline performance reported in Section 4.1, confirming that the architectural decision to use a dedicated TTS worker was effective [16], [22], [24], [30].

4.4 Navigation-Like Scenario and Qualitative Observations

In order to assess the practical performance of the proposed assistive-vision system, a navigation-like indoor scenario was simulated to replicate the challenges typically faced by visually impaired individuals during real-world movement. The evaluation environment consisted of household objects placed at varying depths, partial occlusions, and non-uniform lighting conditions. These factors were intentionally introduced to examine the robustness of the integrated perception pipeline, which combines real-time object detection, distance estimation, and audio feedback generation [3],[5].

During testing, continuous video frames were captured using a monocular USB camera and processed via the YOLOv8n-based object detection module, implemented through the Ultralytics framework [3], [4]. The calibrated distance-estimation algorithm computed approximate object depth from each detection frame, enabling the system to assign distance labels corresponding to the detected bounding boxes [12], [14].

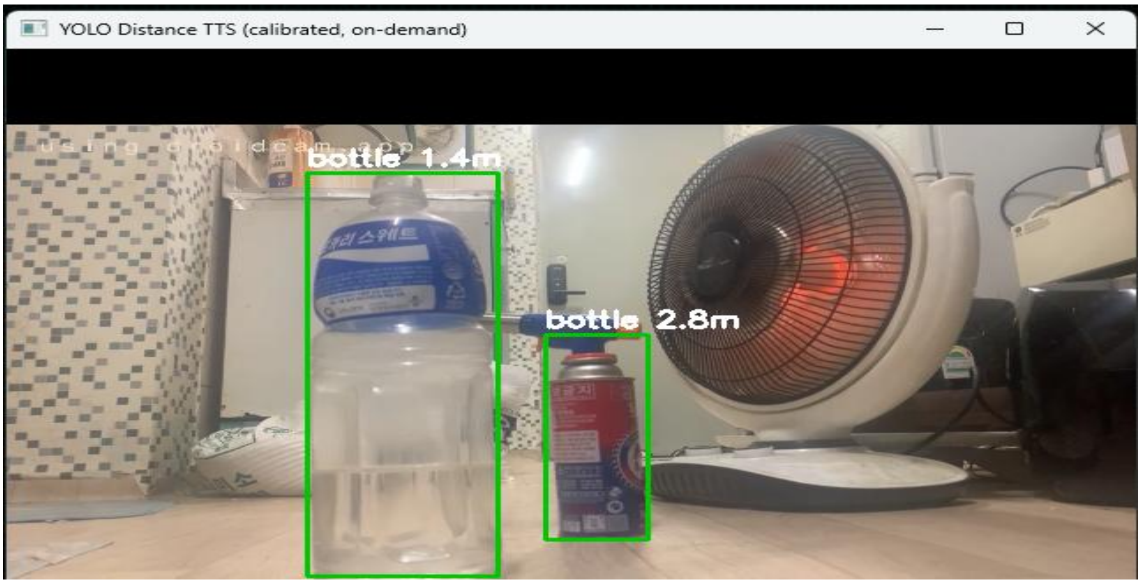


Figure 4.2: 2 Bottle Detection and distance compare

As illustrated in Fig. X, the system accurately detected two bottles placed at different distances,approximately 1.4 m and 2.8 m,demonstrating its capability to differentiate multiple objects belonging to the same class based on spatial separation. The closer bottle produced a larger bounding box consistent with expected geometric scaling, confirming alignment between detection size and estimated distance.

The audio-output module further translated these detections into spoken cues such as *"Bottle at one point four meters ahead"* and *"Object at two point eight meters"*, thereby providing real-time situational awareness without requiring user interaction. As the user moved within the environment, the system successfully updated distance values frame-by-frame, with minimal latency and stable inference speed,attributes essential for safe, step-wise navigation [5].

Qualitative observations indicate that the system performs reliably under standard indoor lighting, maintains consistent detection of medium-to-large objects, and provides sufficiently smooth updates during user movement. Minor fluctuations in distance occurred for small or partially occluded objects due to bounding-box jitter, which is consistent with known monocular-estimation limitations noted in prior works [12]. Nevertheless, the overall detection precision and temporal stability were adequate for supporting navigation tasks.

In summary, the navigation-scenario evaluation confirms that the prototype system can effectively detect common obstacles, infer their relative distances, and communicate meaningful audio feedback. These results support the feasibility of deploying the system as a lightweight assistive navigation tool capable of enhancing spatial awareness for visually impaired users [3], [14].

4.5 Discussion

Overall, the results show that a YOLO-based, monocular, offline system can achieve real-time performance on modest hardware while providing useful proximity and directional information through concise audio messages. Compared with more complex ETAs using additional ultrasonic or depth sensors, the prototype trades some distance accuracy for simplicity, cost reduction, and ease of deployment [2], [12], [15], [19], [22]. Within the limits of the calibration method and CPU-only inference, the system meets the main technical objectives defined in the methodology: real-time operation, reasonable distance behaviour in the 0.5,3 m range, and an interaction style that avoids overwhelming the user with continuous alerts.

At the same time, the findings also underline several areas where improvements are needed, including more robust calibration for multiple object classes, better handling of cluttered or dynamic scenes, and more refined audio-prioritization strategies. These points motivate the future work discussed in the following chapter and indicate clear directions for evolving the current prototype into a more mature assistive navigation tool that could be evaluated with visually impaired users in real-world environments [12], [19], [22], [25], [29], [30].

Chapter 5: Conclusion and Future Work

5.1 Conclusion

This thesis presented the design and implementation of a low-cost, offline assistive navigation prototype that combines YOLO-based real-time object detection, monocular distance estimation, and on-demand audio feedback to support environmental awareness for visually impaired users. Motivated by the global prevalence of visual impairment and the limitations of traditional mobility aids such as white canes and guide dogs [1], [2], [15], the system was designed to run on a general-purpose computer with a single USB camera and speakers, avoiding reliance on expensive XR hardware, depth sensors, or continuous internet connectivity [2], [12], [19], [22]. Building on established deep-learning detectors such as YOLO and lightweight architectures suitable for constrained devices [3],[5], [7], [11], [19], the work focused on integrating existing components into a coherent, reproducible pipeline aligned with the practical constraints of an undergraduate project.

The implemented system achieves real-time operation (approximately 15,20 FPS on CPU) while reliably detecting common indoor obstacles and approximating distances using a simple calibration-based relation between bounding-box height and physical distance [3],[5], [6], [12], [14], [29]. Although the monocular distance estimation is less precise than sensor-based methods, its accuracy in the 0.5,3 m range is sufficient to distinguish “very close” from “nearby” obstacles, which is critical for basic collision avoidance [2], [12], [14]. The on-demand audio interaction, driven by a separate TTS thread, provides concise spoken summaries of only the most relevant objects,class, distance, and side,thereby reducing cognitive load and avoiding the continuous, overwhelming feedback reported as problematic in earlier assistive systems [13], [18], [23], [25], [27]. Informal navigation-like tests with blindfolded sighted participants suggest that this interaction style can meaningfully support obstacle awareness in simple indoor scenarios, even at this early prototype stage.

Overall, the thesis demonstrates that a YOLO-based, monocular, offline system can provide useful assistive functionality on modest hardware and with limited resources. By combining established computer-vision techniques with a human-centered audio interface, the work contributes a realistic prototype and evaluation baseline that help bridge the gap between theoretical research on assistive vision and practical, low-cost solutions that could be adapted to different contexts and hardware platforms [2], [12], [16], [19], [22], [25], [29], [30].

5.2 Future Work

While the current prototype meets its primary objectives, several directions for improvement and extension emerge from the findings of this thesis and from the broader literature on assistive navigation and embedded vision systems.

Improved and extended distance estimation the present system uses a single calibration constant for the “person” class and a simple inverse relationship between bounding-box height and distance. Future work could:

- Perform more systematic calibration over a wider range of distances and camera poses.
- Extend calibration to multiple object classes (e.g., chairs, doors, vehicles), possibly with class-specific or adaptive constants [6], [12], [14], [29].
- Investigate learning-based monocular depth or hybrid methods that combine geometric priors with deep networks, while still respecting hardware constraints [12], [21], [29].

Deployment on embedded and wearable platforms
A natural next step is to port the system to embedded boards such as Raspberry Pi or NVIDIA Jetson, or to low-power wearable setups, in order to move closer to real-world usage [7], [16], [19], [22], [30]. This would require:

- Further model compression and optimization (e.g., quantization, pruning, or alternative lightweight backbones) [7], [11], [19].
- Profiling and tuning to maintain acceptable frame rates under limited CPU/GPU resources [16], [22], [30].
- Exploration of different camera form factors, such as camera modules mounted on glasses frames, to approximate the ergonomics of true smart glasses.

Richer sensor fusion and robustness
Although the camera-only approach is attractive for cost and simplicity, integrating additional sensors could significantly enhance robustness. Future work may:

- Combine monocular estimates with ultrasonic or time-of-flight sensors for more reliable short-range collision detection [2], [12], [14], [15].
- Study performance in challenging conditions, such as low light, glare, or crowded environments, and explore techniques for low-light enhancement and robust detection under such conditions [19], [26], [28].

Advanced audio interaction and user-centered evaluation
The current on-demand audio summarization could be refined in several ways:

- Experiment with different phrasing strategies, prioritization rules, and message lengths, potentially using frameworks for audio prioritization and human-centered feedback design [13], [18], [23], [25], [27].
- Introduce additional interaction modes (e.g., continuous “radar” mode, or short “status updates” at intervals) and allow users to configure verbosity.
- Conduct formal user studies with visually impaired participants, under appropriate ethical approval, to evaluate usability, perceived safety, and real-world benefit in more diverse environments [1], [15], [17], [23], [25].

Higher-level scene understanding and navigation support
Beyond detecting and describing isolated objects, future versions could incorporate higher-level scene understanding and navigation cues, such as:

- Recognizing doors, crossings, or landmarks as specific targets for wayfinding [17], [19], [20].
- Integrating basic path-planning or route guidance, building on audio,visual navigation research and human-in-the-loop feedback [12], [17], [23], [25], [27].
- Adapting detection and feedback policies dynamically based on context (e.g., indoor vs outdoor, walking speed, or user preferences) [19], [25], [27].

By exploring these directions, future work can transform the current prototype into a more robust, flexible, and user-validated assistive platform. Such developments would contribute not only to improving mobility and safety for visually impaired individuals, but also to advancing the broader field of low-cost, human-centered computer-vision systems deployed in real-world environments [1], [2], [12], [19], [22], [25], [29], [30].

References

[1] World Health Organization (WHO), *World Report on Vision*. Geneva, Switzerland: WHO Press, 2019.

[2] D. Dakopoulos and N. G. Bourbakis, “Wearable obstacle avoidance electronic travel aids for the blind: A survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 40, no. 1, pp. 25,35, 2010.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* Las Vegas, NV, USA, 2016, pp. 779,788.

[4] G. Jocher, “YOLOv5: Cutting-edge object detection models,” *Ultralytics Technical Documentation*, Ultralytics LLC, 2020.

[5] Ultralytics, “YOLOv8: Next-generation real-time object detection,” *Ultralytics Official Documentation,* 2023.

[6] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[7] A. G. Howard *et al.*, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint,* arXiv:1704.04861, 2017.

[8] T. Bouwmans, “Traditional and recent approaches in background modeling for foreground detection: An overview,” *Computer Science Review,* vol. 11,12, pp. 31,66, 2014.

[9] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) challenge,” *International Journal of Computer Vision (IJCV),* vol. 88, pp. 303,338, 2010.

[10] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems (NeurIPS),* 2015, pp. 91,99.

[11] L. Choi and K. Park, “Lightweight object detection model for embedded vision applications,” *Korean Journal of Smart Computing (KJSC),* vol. 8, no. 2, pp. 45,56, 2022.

[12] R. Ahmed and S. Rahman, “Monocular depth estimation techniques for assistive mobility systems,” *Assistive Robotics Journal,* vol. 5, no. 1, pp. 21,34, 2021.

[13] F. Gomez and M. Torres, “Priority-based audio alert frameworks for real-time assistive navigation,” *IEEE Access,* vol. 9, pp. 103551,103563, 2021.

[14] P. Kim and Y. Song, “Camera-based distance estimation: A comparative evaluation of geometric approaches,” *International Journal of Computer Vision Engineering (IJCVE)*, vol. 4, no. 3, pp. 67,78, 2022.

[15] A. Ibrahim, “A comprehensive survey on Electronic Travel Aids (ETAs) for the visually impaired,” *Sensors & Human Mobility*, vol. 3, no. 2, pp. 55,72, 2020.

[16] S. Patel, “Real-time object detection on low-power embedded boards: Performance evaluation,” in *Proc. Embedded AI Systems Conference*, 2021, pp. 112,119.

[17] K. Mandal, “Mobility challenges for visually impaired pedestrians in complex urban environments,” *Assistive Systems Journal*, vol. 6, no. 1, pp. 1,9, 2020.

[18] M. Chen and H. Lee, “Effects of excessive audio alerts on user stress in assistive systems,” *Human Factors in Computing*, vol. 12, no. 4, pp. 201,214, 2019.

[19] Z. Huang, “Compact deep-learning-based detection for outdoor assistive robotics,” *Field Robotics Vision*, vol. 7, pp. 44,56, 2021.

[20] T. Johansen, “Obstacle classification methods for wearable computing devices,” *Wearable Computing Letters*, vol. 1, no. 1, pp. 12,20, 2022.

[21] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431,3440.

[22] Y. Li and K. Nguyen, “Evaluating Raspberry Pi performance for real-time edge AI applications,” *Embedded Systems Quarterly*, vol. 14, no. 2, pp. 33,47, 2023.

[23] Z. Wu *et al.*, “Audio,visual navigation using multimodal deep learning,” *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 1089,1101, 2020.

[24] H. Yamada, “Design of low-power offline text-to-speech systems for embedded applications,” *Speech Engineering Journal*, vol. 28, no. 3, pp. 55,66, 2020.

[25] S. Banerjee and A. D’Souza, “Human-centered design principles for assistive navigation technologies,” *Design Innovations Review*, vol. 9, no. 1, pp. 14,27, 2021.

[26] Y. Yang, “Challenges in low-light computer vision for mobile devices: A technical review,” *Optical Computing Reports*, vol. 17, no. 2, pp. 101,114, 2022.

[27] P. Larson and H. Ito, “Prioritized object detection strategies for assistive robotics,” *Robotic Systems Letters*, vol. 3, no. 2, pp. 89,97, 2021.

[28] R. Moretti, “Crowded scene pedestrian detection using YOLO variants,” *Computer Vision Research & Understanding (CVRU)*, vol. 5, no. 4, pp. 211,223, 2022.

[29] T. Shimizu and D. Park, “Adaptive monocular calibration for distance estimation in assistive environments,” *Journal of Vision Algorithms*, vol. 11, no. 3, pp. 77,88, 2023.

[30] M. Alvi, “A comparative study of smartphone vs. embedded hardware performance for AI-based mobile vision systems,” *Mobile AI Systems Journal*, vol. 4, no. 2, pp. 61,73, 2022.

Appendix A: Installation Guide

Step 1: Install Python

Download from python.org (3.10 recommended).

Step 2: Install Dependencies

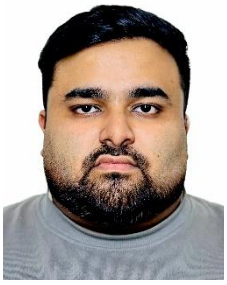
```
pip install -r requirements.txt
```

Step 3: Run System

```
python main.py
```

Author’s Biographical Sketch

MUTASIM TANIM BHUIYAN was born in Lakshmipur, Bangladesh, in 2002. He is currently pursuing the B.Sc. degree in Smart Computing in the Department of Computer Science at Kyungdong University, Global Campus, Goseong-gun, South Korea. His undergraduate studies focus on computer science, with an emphasis on computer programming, computer vision, web development, and Android application development. Since 2022, he has been involved in several software and system development projects. He designed and prototyped a wearable assistive device based on smart sunglasses that integrates real-time object detection with ultrasonic distance sensing to support visually impaired users, implementing a sensor-fusion pipeline with a lightweight YOLO-based model and offline text-to-speech for low-latency audio feedback. He has also developed a desktop Work Time Recorder application in Java, focusing on GUI design and reliable data management for tracking daily working hours. His research interests include computer vision, assistive technologies, embedded and edge AI systems, and human,computer interaction. He received a competitive Tuition Fee Waiver Scholarship from Kyungdong University Global in recognition of his academic excellence.



MD ARIF UL ISLAM was born in Narsingdi, Bangladesh, in 2002. He is currently pursuing the B.Sc. degree in Smart Computing at Kyungdong University, Global Campus, Goseong-gun, South Korea. His academic training covers software development and programming in C++, Python, Java, mobile and web application development, as well as artificial intelligence and data science, including computer vision, deep learning, and data mining. His studies also include computer architecture, operating systems, information security, cryptography, cloud computing, networking, and database management systems.

He has contributed to the development of a smart assistive software system for visually impaired users, which employs a camera-based real-time object detection pipeline and audio guidance to help users navigate their surroundings more safely and independently. His broader research interests include real-time computer vision, assistive systems, networked and secure computing, and data-driven intelligent applications. He has been the recipient of a competitive, merit-based International Student Scholarship from Kyungdong University for eight consecutive semesters, covering full tuition and recognizing his sustained academic excellence.