|  |
| --- |
| C:\Documents and Settings\Mohammad Alzubaidi\My Documents\YU\logo2.jpg  **Graduation Project Report**  **SightOfBlind: An AI-Powered Voice Navigation Assistant**  **Momen Mahmoud Bani-Ali 2021905109**  **Anas Emad Falah Obeidat 2021905018**  **Omar Mahmoud Al-Rashdan 2021905092**  **Supervisor: Dr. Enas Alikhashashneh**  **Semester: Second 2024/2025**  **Date: 18 May 2025** |

**Yarmouk University**

**Faculty of Computer Science and Information Technology Department of Information Systems**

# Students' Property Right Declaration and Anti-Plagiarism Statement

We hereby declare that the work in this graduation project at Yarmouk University is our own except for quotations and summaries which have been duly acknowledged. This work has not been accepted for any degree and is not concurrently submitted for award of other degrees. It is the sole property of Yarmouk University, and it is protected under the intellectual property right laws and conventions.

We hereby declare that this report is our own work except from properly referenced quotations and contains no plagiarism.

We have read and understood the school's rules on assessment offences, which are available in the Yarmouk University Handbook.

|  |  |  |
| --- | --- | --- |
| **Students:** | |  |
| Name: Anas Obeidat | Signature: **Obeidatt** | Date: **18 May 2025** |
| Name: Momen Bani-Ali | Signature: **Momen** | Date: **18 May 2025** |
| Name: Omar Al-Rashdan | Signature: **Omar** | Date: **18 May 2025** |
| Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Table of Contents

[Students' Property Right Declaration and Anti-Plagiarism Statement i](#_Toc199367457)

[List of Figures iv](#_Toc199367458)

[Abstract v](#_Toc199367459)

[Chapter 1: Introduction 1](#_Toc199367460)

[Chapter 2: Problem Statement 5](#_Toc199367461)

[Chapter 3: project objectives 7](#_Toc199367462)

[Chapter 4: Literature Review 8](#_Toc199367463)

[Chapter 5: Methodology 10](#_Toc199367464)

[Chapter 6: Result and Discussion 23](#_Toc199367465)

[Chapter 7: Limitations 27](#_Toc199367466)

[Chapter 8: Conclusion 29](#_Toc199367467)

[Chapter 9: Future Work 31](#_Toc199367473)

[References 33](#_Toc199367474)

# List of Figures

[Figure 1 : Label distribution in the custom YOLO dataset 11](#_Toc199367611)

[Figure 2 : Heatmaps of object center positions (left) and bounding box sizes (right) in the dataset. 11](#_Toc199367612)

[Figure 3 : YOLO object detection Workflow 14](#_Toc199367613)

[Figure 4 : High level YOLOv11 architecture 15](#_Toc199367614)

[Figure 5 : Raspberry pi 19](#_Toc199367615)

[Figure 6 : Webcam 19](#_Toc199367616)

[Figure 7 : Airpods 20](#_Toc199367617)

[Figure 8 : Cooling System and Case 20](#_Toc199367618)

[Figure 9 : Power Bank 21](#_Toc199367619)

[Figure 10 : Final Product 22](#_Toc199367620)

[Figure 11 : YOLOv11 training and validation curves 24](#_Toc199367621)

[Figure 12 : Normalized confusion matrix 25](#_Toc199367622)

# Abstract

Independent navigation remains a major challenge for visually impaired and blind individuals, significantly impacting their autonomy and quality of life. This project introduces an AI-powered assistive system embedded in a wearable hat, designed to enhance spatial awareness and guide users toward specific objects or locations. Powered by a Raspberry Pi, the system leverages deep learning for object detection and integrates voice interaction through speech-to-text (STT) and text-to-speech (TTS) technologies.

Users can issue voice commands—such as “find the keys” or “where is the exit”—which are interpreted via STT and processed by the system. Using a predefined map of rooms and objects, the system either determines the current location or directly searches for the requested object within the environment. Upon detecting the target, it calculates its direction relative to the user (e.g., “forward,” “to your left,” “to the southwest”) and communicates guidance through AirPods using TTS feedback.

By combining real-time object detection, spatial context awareness, and seamless voice interaction on accessible hardware, the system aims to improve independent mobility and situational understanding for visually impaired users. Ongoing development focuses on enhancing detection accuracy, robust room localization, and natural voice responsiveness across diverse real-world conditions.

Future work in this area could involve expanding voice command support to work with the Arabic language, allowing the system to serve a wider user base. The model will also be updated to recognize a larger set of objects and understand their statuses, for example, distinguishing between an open and closed door. Incorporating depth sensors for better distance estimation, improving low-light detection capabilities, and optimizing power consumption to support longer use in portable scenarios.

**Keywords**— Visually Impaired, YOLO11, Raspberry Pi, STT, TTS, Deep learning, Deep Learning, Assistive Technology.

# Chapter 1: Introduction

**1.1 Introduction and Background**

For many of us, finding the door, spotting where we left our keys, or navigating a hallway is second nature. But for visually impaired individuals, these simple actions can become daily challenges that limit independence and confidence. While tools like canes and guide dogs help to some extent, they often fall short in providing information about what is around and where it is. That missing spatial awareness is something most sighted people don’t even think about—until it’s gone.

We were motivated to work on this project after realizing how much of our world relies on visual interaction. The blind community still lacks affordable, intelligent tools that can help them interact with their environment in a more intuitive way—tools that listen, see, and speak just like we do.

Technology has come a long way. Thanks to advances in artificial intelligence, computer vision, and embedded systems, it’s now possible to build real-time, smart systems that detect objects, understand speech, and respond with helpful feedback. We saw an opportunity to bring these technologies together into one wearable solution.

Our project, SightOfBlind, is a smart hat designed to assist blind users in their daily lives. By combining a camera, deep learning object detection, and voice interaction, the system lets users ask questions like “Where is the door?” or “Find the chair,” and then guides them using audio feedback. It’s like having a helpful friend who can see for you and speak to you.

With this project, we hope not just to build a useful tool—but to take a meaningful step toward making daily navigation more accessible, more human, and more empowering for people who can’t rely on their eyes.

* 1. **Aims and Objectives.**

The main aim of this project is to design and implement an intelligent, wearable assistive system that helps visually impaired individuals detect and locate objects in real-time using voice commands and audio feedback. The goal is to improve indoor navigation, increase autonomy, and reduce reliance on external help.

To achieve this aim, the project focuses on the following objectives:

* Develop a wearable system using Raspberry Pi and a camera to perform real-time object detection.
* Enable users to give voice commands using speech-to-text (STT) to search for specific objects.
* Provide direction-based voice feedback using text-to-speech (TTS) technology.
* Integrate lightweight AI models suitable for real-time performance on embedded devices.
* Ensure the system operates smoothly in indoor environments with limited lighting or varying layouts.
* Test and evaluate the system’s accuracy, latency, and usability with different object types and distances.
  1. **Solutions History**

Over the years, several technologies have been developed to assist the visually impaired in navigating their surroundings. The most common tools include white canes and guide dogs, which are effective for detecting obstacles but do not provide directional awareness or object identification capabilities.

With the advancement of technology, electronic aids such as ultrasonic-based obstacle detectors, GPS navigation apps, and wearable sensors have been introduced. Some commercial systems offer voice guidance or alert users to nearby objects, while research projects have experimented with smart glasses, infrared sensors, and mobile applications that use computer vision to describe scenes.

However, many of these solutions have significant limitations. Some are limited to outdoor use and rely on GPS signals, which are not available indoors. Others require high-cost hardware or are too bulky to be practical. Most importantly, few systems allow the user to issue custom voice- commands to locate specific objects, and even fewer integrate real-time object detection with natural voice feedback in a lightweight and wearable form.

These limitations leave a clear gap in accessible, real-time, and voice-interactive systems tailored for indoor use — a gap that SightOfBlind aims to fill.

* 1. **Main Solution Idea**

SightOfBlind proposes a wearable AI-powered system designed to assist visually impaired users in locating objects and navigating indoor environments. The solution is built around a compact, smart hat equipped with a Raspberry Pi, a camera, and audio output (AirPods or earphones). The user interacts with the system through voice commands, making it hands-free and intuitive.

When a user issues a command such as “find the chair” or “where is the door?”, the system converts the speech into text, processes it to identify the target object, and then uses a pre-trained object detection model to locate that object in the camera feed. Once detected, the system calculates the object's relative direction (e.g., in front, to the left, etc.) and gives clear audio feedback to the user using text-to-speech.

The design focuses on real-time performance, minimal hardware, and ease of use. By integrating speech-to-text (STT), object detection, and text-to-speech (TTS) into one lightweight wearable, SightOfBlind empowers users to explore their surroundings independently, quickly, and safely — without relying on expensive or complex systems.

* 1. **Key Technical Details:**

The SightOfBlind system combines computer vision, voice interaction, and embedded hardware to deliver an intelligent, wearable assistant for blind users. Below is a summary of the key technical components that power the system:

**Hardware Components:**

* **Raspberry Pi 5 (8GB RAM)**  
  Serves as the main processing unit, selected for its improved performance and memory, capable of handling real-time AI inference.
* **USB Webcam**  
  Used instead of the Pi Camera Module to simplify setup and avoid complex driver configurations, while still providing high-resolution image input.
* **AirPods Microphone**  
  Utilized for capturing clear voice commands from the user, offering mobility and convenience.
* **Bluetooth Audio Output (AirPods)**  
  Delivers voice feedback to the user via synthesized speech.
* **Cooling System and Protective Case**  
  Installed to ensure thermal stability and physical protection during extended usage.
* **Portable Power Bank**  
  Enables mobility by supplying power to the Raspberry Pi.

**Software Components:**

* **Python**  
  Used as the core language for system logic, hardware integration, and AI model execution.
* **YOLOv11n (Original + Custom Model)**  
  Combines the original YOLOv11n object detection model with a custom-trained version tailored to detect commonly encountered indoor objects relevant to blind users.
* **OpenCV**  
  Handles real-time image capture, preprocessing, and visual frame handling.
* **SpeechRecognition**  
  Transforms spoken voice commands into text (speech-to-text) for further processing.
* **GTTS (Google Text-to-Speech)**  
  Converts response text into audible speech, providing real-time verbal feedback to the user.
* **Custom Directional Logic**  
  Analyzes the detected object's position in the image and determines its direction relative to the user (e.g., "in front", "to your right").

The entire system operates in real time with optimized performance to ensure low latency and user-friendly feedback, all running on a compact and portable setup.

# Chapter 2: Problem Statement

Visually impaired individuals often face significant challenges in their daily lives, particularly when it comes to navigating and interacting with their surroundings without assistance. Everyday tasks that most people take for granted—such as finding a door, locating a chair, or simply moving from one room to another—can become stressful, time-consuming, and even dangerous for someone who cannot see. The lack of visual input limits their ability to understand spatial layout, recognize obstacles, or identify objects, especially in unfamiliar indoor environments. This results in a heavy reliance on caregivers or physical aids, which can affect their confidence, mobility, and sense of independence.

Traditional mobility aids like white canes and guide dogs are undoubtedly useful but come with limitations. While they help detect immediate physical obstacles, they offer no information about what those obstacles are, where specific objects are located, or how to reach them. They also do not support interaction through speech or provide any form of directional awareness beyond tactile feedback. This creates a gap between what the user needs to know and what the aid is capable of delivering.

In response to these challenges, various technological solutions have been introduced over the years, including GPS-based mobile apps, ultrasonic sensors, infrared-based navigation tools, and wearable devices equipped with audio alerts. Some of these systems can guide users to destinations or alert them to the presence of obstacles. However, most of them are primarily designed for outdoor navigation and lose functionality indoors where GPS signals are unreliable or absent. Moreover, these systems often require complex calibration, are cost-prohibitive, or fail to provide a truly seamless and interactive experience. Few of them allow users to issue personalized voice commands or dynamically search for specific objects in real time.

Another critical limitation of existing technologies is their lack of adaptability and real-time responsiveness. For instance, a user might want to say “find the chair” or “where is the bag,” but most systems do not support object-specific queries or natural language commands. Additionally, many existing models are too computationally heavy to run on small embedded devices, making them impractical for portable, wearable use.

To bridge these gaps, there is a pressing need for a smart, lightweight, and real-time system that combines advanced object detection with voice interaction in a wearable form factor. Such a system should be capable of understanding spoken commands, detecting and identifying objects in the user’s field of view, and providing verbal guidance about their direction and location—all in real time and without internet dependency.

The SightOfBlind project was developed to address this exact need. By integrating modern AI techniques, including computer vision, into a compact wearable system powered by Raspberry Pi 5, this project introduces a practical solution tailored for visually impaired individuals. The system allows users to speak naturally, search for specific objects, and receive immediate spoken feedback that helps them orient themselves in their environment. This not only enhances user independence but also contributes to safer, more confident navigation in daily life.

# Chapter 3: project objectives

The aim of the SightOfBlind project is to develop a wearable assistive system that enhances the spatial awareness and autonomy of visually impaired individuals by enabling them to locate and identify objects through voice interaction and real-time visual detection.

To fulfill this aim, the project sets out the following specific objectives:

* Design a wearable and lightweight hardware setup that includes a Raspberry Pi 5, a webcam, microphone, cooling system, and power source, all integrated into a portable form suitable for daily indoor use.
* Implement a real-time object detection module using the YOLOv11n model, combining both the original version and a custom-trained version to detect a set of commonly encountered indoor objects with high accuracy.
* Integrate speech-to-text (STT) functionality to allow users to issue voice commands such as “find the chair” or “where is the bag,” using AirPods or any compatible microphone.
* Process voice commands and interpret the target object from the user’s input to trigger the detection process.
* Determine the spatial direction of the detected object relative to the user (e.g., front, left, right) and deliver audio guidance through Bluetooth audio output using Google Text-to-Speech (gTTS).
* Test and evaluate the system in different indoor environments to measure usability, detection accuracy, response time, and overall user experience.

By meeting these objectives, SightOfBlind aims to provide a practical, intelligent, and user-friendly solution that empowers visually impaired users to navigate more confidently and independently in their daily lives.

# Chapter 4: Literature Review

The development of assistive technologies for visually impaired individuals has been a growing field of research and innovation, aiming to enhance independence, mobility, and spatial awareness. A wide variety of tools and systems have been explored — ranging from traditional aids like white canes to advanced computer vision-based wearable devices — each with its own strengths and limitations.

Traditional navigation tools such as white canes and guide dogs remain the most widely used. They help users detect immediate obstacles through touch or movement guidance but offer no information about the type of objects around or their relative direction. Moreover, these aids lack interaction features such as voice communication or real-time feedback [[1]](#ref1).

To extend these capabilities, early Electronic Travel Aids (ETAs) were developed using ultrasonic or infrared sensors to detect obstacles and convert distance information into vibrations or sound cues. While helpful, these devices provided only basic proximity warnings without contextual or semantic understanding of the environment [[2].](#ref2)

With the rise of mobile computing, smartphone applications such as **Seeing AI** and **Be My Eyes** emerged, offering AI-powered scene description, text reading, and facial recognition. These apps enable users to gain more awareness of their environment through the phone’s camera and cloud-based AI models [[3]](#ref3). However, most of these apps require a stable internet connection, are not wearable, and are limited in their ability to perform continuous real-time detection indoors.

In the realm of artificial intelligence, object detection models such as **YOLO (You Only Look Once)** have been widely used for real-time visual recognition. These models are capable of detecting multiple objects in a single frame with high speed and accuracy. The most recent version, **YOLOv11**, introduces transformer-enhanced feature extraction and lightweight optimization, making it suitable for edge computing on embedded devices like the Raspberry Pi 5 [[4]](#ref4)[[5]](#ref5).

Speech recognition and text-to-speech technologies have also become more reliable and accessible in recent years. Open-source tools like **SpeechRecognition** and **Google Text-to-Speech (gTTS)** allow systems to interpret user voice input and deliver synthesized audio responses. These capabilities are especially valuable in hands-free systems, allowing users to interact with devices naturally and efficiently [[6]](#ref6).

Several recent projects and commercial systems share conceptual similarities with *SightOfBlind*. For instance, **OrCam MyEye** is a wearable device that attaches to glasses and uses a camera to read text, recognize faces, and provide audio feedback. While it offers offline functionality and portability, it is prohibitively expensive and does not support voice-command-based object searching [[7]](#ref7).

Similarly, **Seeing AI** by Microsoft provides advanced scene analysis and narration through a smartphone camera, but it is not wearable and depends heavily on cloud processing [[3]](#ref3).

Another innovative example is the **AI Suitcase**, developed by IBM Japan and academic partners, which guides blind users using computer vision, LiDAR, and voice. However, it is bulky and still in the research phase, lacking portability and affordability [[8]](#ref8).

Despite these advancements, a gap still exists in the availability of low-cost, wearable, offline-capable assistive systems that support custom object detection and respond to natural voice commands in real time. Many current solutions either require internet connectivity, offer only limited interaction, or are not designed for continuous indoor use.

The *SightOfBlind* project addresses this gap by combining a compact Raspberry Pi 5 setup with a webcam, YOLOv11-based object detection, and offline voice interaction. The system allows users to speak naturally, search for specific objects, and receive directional voice feedback, enabling visually impaired users to navigate their environment more independently and with greater confidence.

# Chapter 5: Methodology

**5.1.1 Data Collection**

The effectiveness of the SightOfBlind system relies heavily on the quality and diversity of data used to train the object detection model. To ensure accurate identification of indoor objects relevant to visually impaired users, a combination of pre-existing datasets and a custom-labeled dataset was used.

Initially, the project leveraged the COCO (Common Objects in Context) dataset, which contains over 330,000 images across 80 object categories commonly found in real-world scenes. This dataset provided the foundation for object detection using the original YOLOv11n model.

While COCO includes useful classes such as "chair," "couch," and "bottle" it lacks focus on specific object types that are important in indoor navigation scenarios for blind users.

To address this gap, we created a custom dataset by collecting images of frequently encountered objects in indoor environments such as:

Door, Fire Extinguisher, Gas Stove, Key, Light Switch, Mirror, Shower, Stairs, Wallet, Washing Machine, Water Cooler, Window, table lamp and trash can.

These images were gathered from real indoor spaces (offices, classrooms, and home environments) and augmented using rotation, brightness variation, flipping, shearing, adding noise and scaling to improve model generalization. All images were manually annotated using Roboflow to generate bounding boxes and label files in YOLO format.

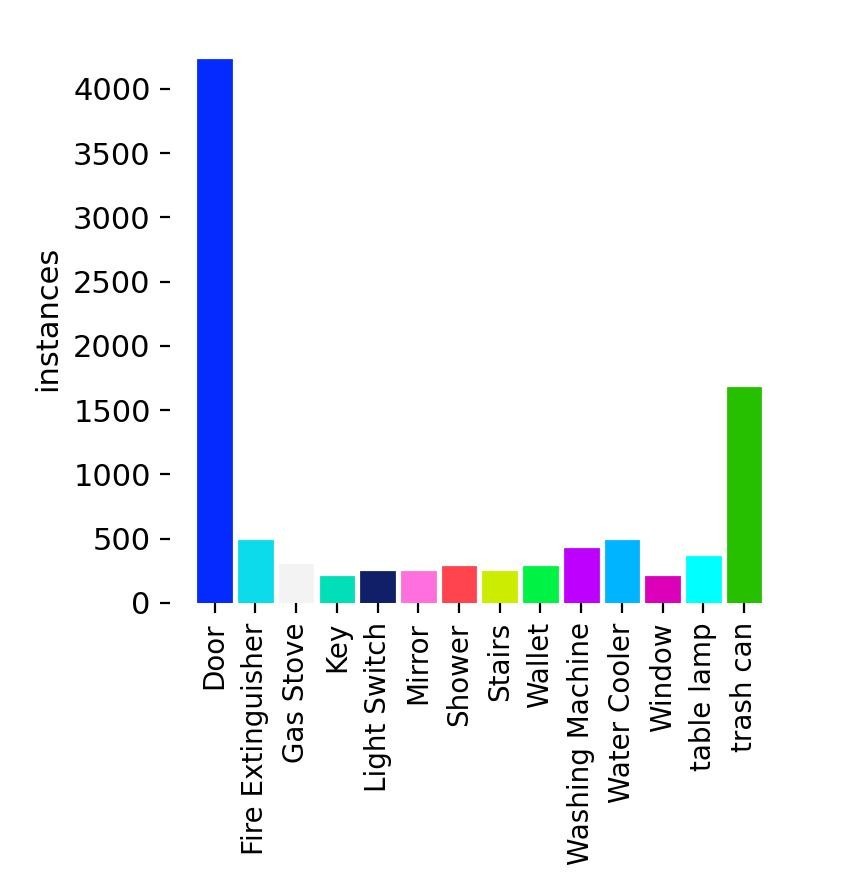
In Fig 1 This bar chart shows how many times each object appears in the custom YOLO dataset. The **"Door"** class has the most instances (over 4000), followed by **"Trash Can"** with around 1500. Other objects like **Fire Extinguisher**, **Mirror**, and **Shower** appear much less often.

Figure 1 : Label distribution in the custom YOLO dataset

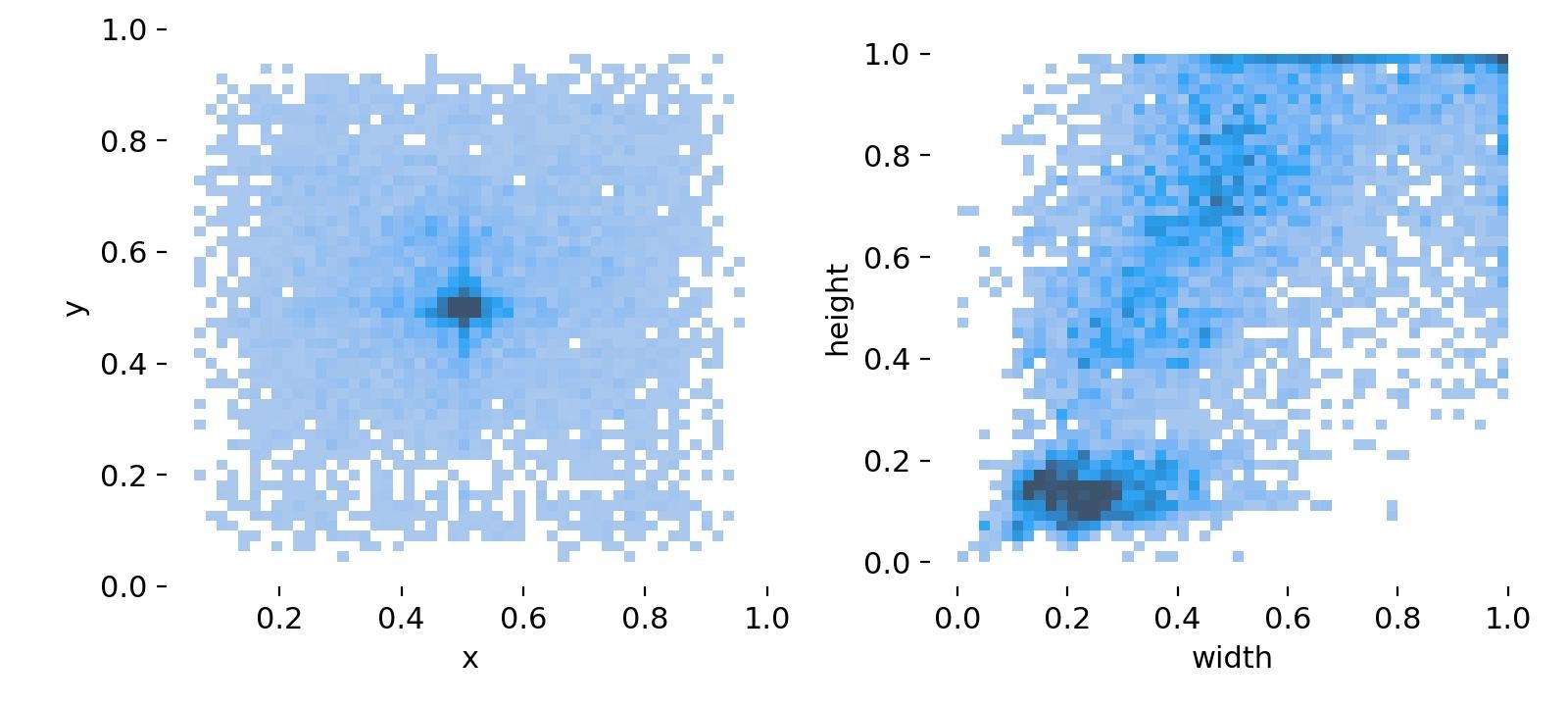
In Fig 2 . The left plot shows where object centers (x, y) are usually found — mostly near the center. The right plot shows the width and height of objects — most of them are small in size. This helps understand object placement and size for model training.

Figure 2 : Heatmaps of object center positions (left) and bounding box sizes (right) in the dataset.

In total, the training dataset included:

2,000+ images for core object classes,

Divided into training (80%), validation (10%), and testing (10%) splits.

This hybrid approach—using both a large-scale general-purpose dataset and a custom, task-specific dataset—ensured the model would generalize well while also accurately recognizing critical objects in real-world deployment environments.

**5.1.2 Voice Command Examples**

For the speech component, no custom dataset was collected, as the system uses pre-built libraries that leverage on-device speech-to-text (STT) functionality through Google’s SpeechRecognition API. This eliminates the need for training a speech model from scratch.

Instead, the project defined a **list of target commands**, such as:

* “Find the chair”
* “Where is the door?”
* “Locate the bag”

These commands are interpreted as object labels and used to trigger the detection process.

**5.1.3 Audio Output Texts**

The system’s audio output is generated dynamically using gTTS based on the recognized object and its relative direction. No audio recording dataset was required for this phase, as the system does not classify speech — it only generates it.

**5.2 Preprocessing**

Prior to training the object detection model, a series of preprocessing steps were applied to ensure the quality, consistency, and effectiveness of the data used in the SightOfBlind system. These steps were crucial for both the public and custom image datasets.

**5.2.1** **Image Resizing and Normalization**

All input images were resized to 416 × 416 pixels, the optimal input resolution for YOLOv11n. This resizing ensured uniformity across the dataset and compatibility with the model architecture. Images were also normalized by scaling pixel values to the range [0, 1] for faster convergence during training.

**5.2.2 Annotation Conversion**

Annotations created using Roboflow were exported in YOLO format, where each .txt label file contains object class indices along with bounding box coordinates relative to image dimensions. These annotations were structured automatically into training, validation, and test directories.

**5.2.3 Data Augmentation**

To increase dataset variability and improve model generalization, the following augmentation techniques were applied:

* Random horizontal flipping
* Random brightness/contrast adjustments (±23%)
* Rotation (±15 degrees)
* Shear (±12 degrees horizontal, ±12 degrees vertical)
* Noise injection (up to 1.88% pixels)

These augmentations helped simulate real-world scenarios such as varied lighting conditions, angles, and occlusions, making the model more robust in unseen indoor environments.

**5.2.4 Dataset Splitting**

The final dataset was split into:

80% Training Set

10% Validation Set

10% Testing Set

This division ensured sufficient data for learning while preserving unbiased samples for validation and evaluation purposes.

**5.3 Techniques**

The SightOfBlind system is built on a combination of computer vision and voice interaction techniques, designed to work in real-time on embedded hardware. These techniques were carefully selected for their performance, accuracy, and suitability for deployment on resource-constrained devices such as the Raspberry Pi 5.

**5.3.1 Object Detection – YOLOv11n**

The system uses the YOLOv11n (nano) object detection model, which is a lightweight and optimized variant of the YOLOv11 family. YOLO (You Only Look Once) is a state-of-the-art real-time object detection algorithm that processes an entire image in a single pass through the network, making it extremely fast and efficient.

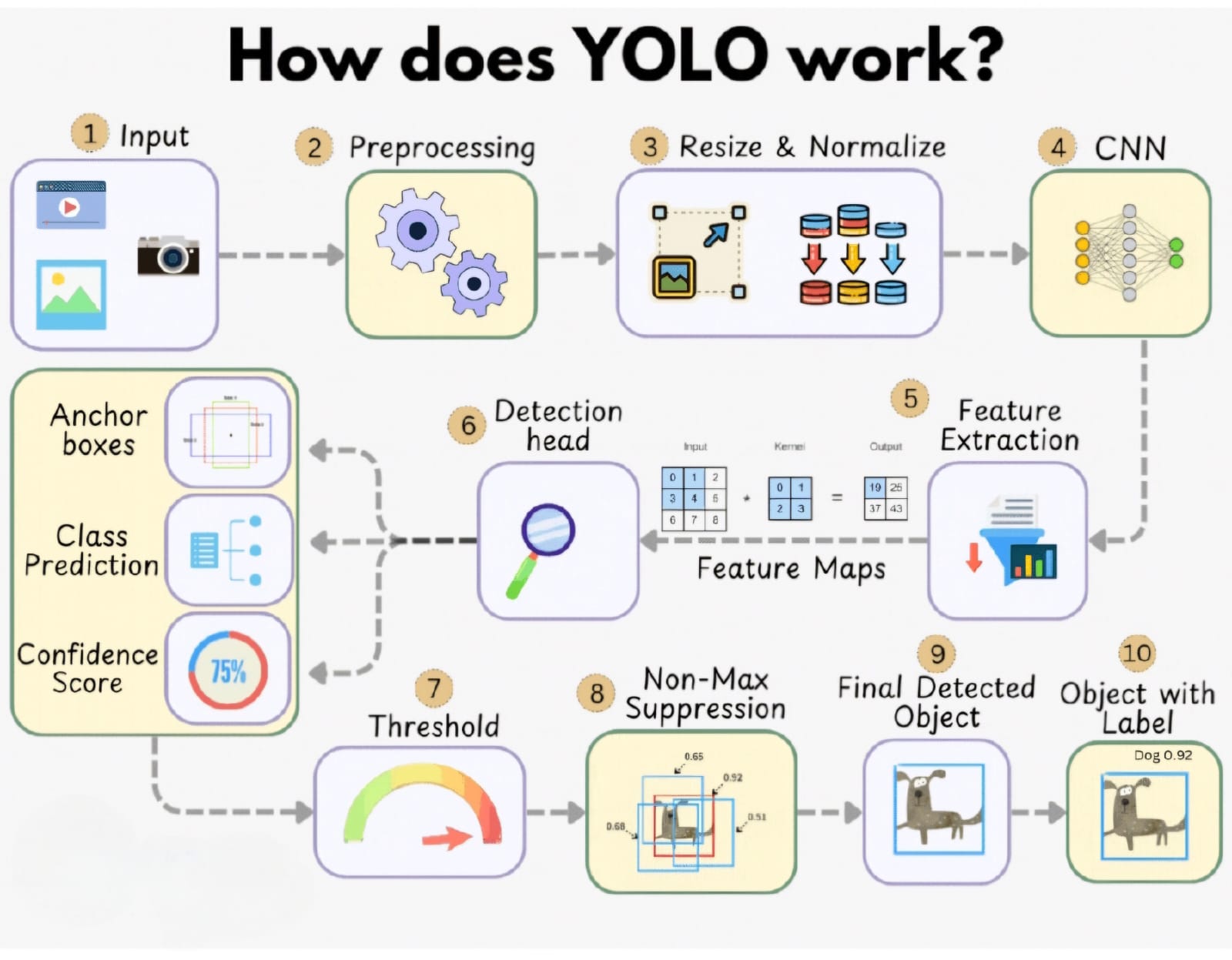
In Fig3. explains the full workflow of the YOLO (You Only Look Once) object detection algorithm. It begins with an input image or video, which is preprocessed, resized, and normalized. The image then passes through a CNN to extract feature maps. These features are used by the detection head to predict bounding boxes, class labels, and confidence scores. After applying a confidence threshold and Non-Max Suppression (NMS), the final objects are detected and labeled.

Figure 3 : YOLO object detection Workflow

In Fig4. provides a detailed view of the **YOLOv11 architecture**, which is divided into three main components:

* **Backbone**: Responsible for feature extraction using convolutional layers and CSP (Cross Stage Partial) blocks. It processes the input image through multiple stages, progressively reducing spatial resolution while increasing feature depth.
* **Neck**: Combines and refines multi-scale features using layers such as SPPF, C2f, and upsampling operations. This part enhances the feature maps for better object localization and classification.
* **Head**: Generates predictions at multiple scales (e.g., P3, P4, P5), including object classification, bounding box coordinates, and confidence scores. It uses anchor-free detection layers to support real-time object detection across various object sizes.

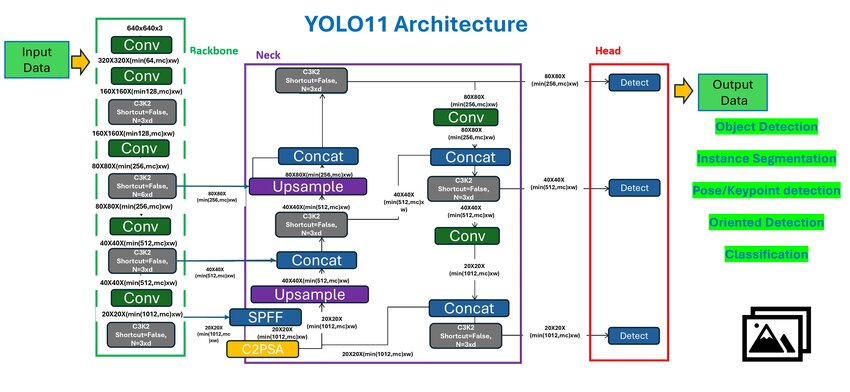
The architecture is optimized for **speed and accuracy**, making it well-suited for edge devices like Raspberry Pi in projects such as *SightOfBlind*.

Figure 4 : High level YOLOv11 architecture

Two models were used:

* The pre-trained YOLOv11n model trained on the COCO dataset for general object classes.
* A custom-trained YOLOv11n model fine-tuned on indoor-specific objects such as chairs, bags, bottles, doors, and tables.

YOLOv11n was selected due to its:

* Fast inference speed suitable for real-time applications,
* Compact size ideal for embedded deployment,
* Strong accuracy even on small objects.
* The model was exported to TFLite format with float16 quantization to reduce memory usage and improve inference speed on the Raspberry Pi 5.

**5.3.2 Voice Interaction – STT and TTS**

To make the system hands-free and accessible, voice-based interaction is implemented through two main components:

**Speech-to-Text (STT):**

The SpeechRecognition Python library is used to capture and convert the user’s voice command into text. This allows the user to say commands such as “find the chair” or “where is the door,” which the system interprets to identify the target object class.

**Text-to-Speech (TTS):**

Once the object is detected and its direction is calculated, the system uses Google Text-to-Speech (gTTS) to convert the response into spoken feedback. This audio is played through Bluetooth earphones (e.g., AirPods), allowing the user to receive real-time, natural guidance.

**5.3.3 Directional Feedback Logic**

After an object is detected, the system calculates its relative position in the frame (left, center, or right) and provides a corresponding verbal response. For example:

“The chair is in front of you.”

“The bag is to your right.”

This logic is implemented by dividing the image into directional zones and interpreting the bounding box center accordingly.

**5.3.4 Integration Architecture**

The entire system is programmed in Python and runs on the Raspberry Pi 5 under a Linux environment. The modules (camera input, object detection, voice command, TTS feedback) run asynchronously for smooth interaction and reduced latency. The webcam continuously captures video, and upon receiving a voice command, the relevant detection task is triggered.

This combination of lightweight vision models, speech processing, and directional logic forms the intelligent core of *SightOfBlind*, allowing it to perform real-time detection and communication efficiently on a portable, wearable device.

**5.4 Performance Metrics**

To assess the effectiveness and reliability of the SightOfBlind system, several performance metrics were used. These metrics were chosen to evaluate both the accuracy of object detection and the real-time usability of the system in indoor environments.

**5.4.1 Object Detection Metrics**

The following standard metrics were used to evaluate the YOLOv11n model on the test dataset:

* **Precision:** Measures how many of the detected objects were correct (true positives / (true positives + false positives)).
* **Recall:** Measures how many actual objects the model correctly identified (true positives / (true positives + false negatives)).
* **mAP (mean Average Precision):** Used to evaluate the overall accuracy across all classes, calculated as the average of precision scores across different Intersection over Union (IoU) thresholds.
* **F1-Score:** The harmonic mean of precision and recall, giving a balanced measure of detection performance.

**5.4.2 Real-Time System Performance**

For a wearable system aimed at real-time guidance, system responsiveness is crucial. The following performance indicators were used during deployment testing on the Raspberry Pi 5:

Frame Rate (FPS):

The system achieved ~8–12 FPS during object detection using the TFLite model with float16 quantization, which is sufficient for responsive interaction in a slow-paced navigation context.

Latency (Command-to-Feedback Time):

Time taken from receiving a voice command to delivering voice feedback:

~1.5–2.5 seconds, depending on detection confidence and direction calculation.

Model Inference Time:

Average time taken for one object detection pass:

~80–100 ms per frame on the Raspberry Pi 5 using YOLOv11n TFLite model.

**5.4.3 Audio Interaction Quality**

STT Accuracy:

Using controlled indoor environments, voice recognition had over 95% accuracy in understanding short commands.

TTS Clarity:

Google Text-to-Speech generated clear, natural-sounding feedback suitable for immediate user understanding.

**5.5 Product**

The final SightOfBlind system is a fully functional, wearable prototype designed to assist visually impaired users in locating objects and navigating indoor environments through voice interaction and real-time object detection. The product integrates hardware and software components into a compact, user-friendly form factor.

**5.5.1 Hardware Overview**

**Raspberry Pi 5 (8GB RAM):**

Serves as the central processing unit, running all AI models and communication modules locally.

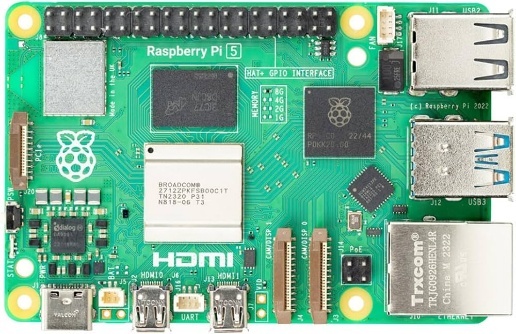


Figure 5 : Raspberry pi

**USB Webcam:**

Mounted on the front of a wearable hat to capture the user's view in real time. Chosen over the official Pi Camera for easier integration and more consistent performance.



Figure 6 : Webcam

**AirPods (Microphone + Audio):**

Act as both the input (microphone for voice commands) and output (audio feedback) interface. This offers hands-free interaction and portability.

Figure 7 : Airpods

**Cooling System and Protective Case:**

Installed to prevent overheating during prolonged usage and to protect the Raspberry Pi from physical damage.

Figure 8 : Cooling System and Case

**Power Bank:**

Provides portable power supply, making the device independent of wall power and suitable for real-world usage.

Figure 9 : Power Bank

**5.5.2 Software Components**

**Object Detection Module (YOLOv11n):**

Runs a quantized TensorFlow Lite version of YOLOv11n, customized to detect specific indoor objects (e.g., chairs, bags, bottles, doors). Triggered upon receiving a user voice command.

**Voice Interaction:**

* SpeechRecognition (STT): Converts spoken commands into text.
* gTTS (Text-to-Speech): Converts detection results and object direction into spoken audio feedback.

**Directional Logic:**

Processes bounding box position to determine if the object is on the left, right, or center, then verbalizes its location relative to the user (e.g., “The chair is to your right”).

**5.5.3 User Interaction Flow**

1. The system continuously listens for a voice command via the AirPods microphone.
2. Upon detecting a command such as “find the bag,” the system uses STT to extract the target object class.
3. The object detection module activates and searches for the specified object in the camera feed.
4. Once the object is detected, the system calculates its position relative to the user.
5. gTTS generates an audio response, such as “The bag is in front of you,” and plays it through the user’s Bluetooth earphones.
   * 1. **Design Considerations**

* **Portability:** All components are integrated into a hat-mounted format.
* **Offline Capability:** The system works entirely offline — no internet is needed for detection or voice interaction.
* **Low Latency:** Average end-to-end command-to-response time is around 2 seconds, ensuring a responsive experience.
* **Cost-Efficiency:** The system is built using affordable components, making it a realistic solution for wider adoption.



Figure 10 : Final Product

The final product offers a seamless blend of AI-powered vision and natural voice interaction, enabling visually impaired users to move more confidently in indoor environments without relying on external assistance.

# Chapter 6: Result and Discussion

**6.1 Results and Discussion**

After completing the design, training, and integration phases, the SightOfBlind system was tested in various real-world indoor environments to evaluate its usability, accuracy, and responsiveness from a functional and human perspective.

The system was deployed on a Raspberry Pi 5 worn in a portable configuration with an attached USB webcam and Bluetooth-connected AirPods. It was used to detect target objects (e.g., “chair”, “bag”, “door”) based on voice commands and provide directional guidance via spoken feedback.

**6.2 Real-World Performance**

While technical metrics (described in Section 5.4.1) showed promising detection accuracy and inference speeds, the true value of SightOfBlind was evident during hands-on testing. Users were able to interact with the system naturally, issuing voice commands and receiving timely, relevant audio responses. This created a seamless experience that required no physical touch, making the interaction more intuitive for visually impaired use.

The system successfully detected and announced the position of various objects in diverse indoor settings — including classrooms, dorm rooms, and hallways — even when objects were partially obstructed or placed at varying angles. While background clutter occasionally confused the model, its custom training allowed it to recover quickly and maintain high usability.

**6.3 User-Centered Observations**

During informal testing sessions (with sighted users blindfolded to simulate the user experience), several observations emerged:

* Voice commands were consistently understood, provided they were spoken clearly and in a quiet environment.
* The audio feedback was easy to follow and helped users orient themselves toward the target object without additional help.
* The wearable form factor (hat-mounted camera and belt-mounted Pi) felt lightweight and didn't obstruct movement.
* This user feedback confirmed that the system was not just technically functional — but actually usable and helpful in practice.

**6.4 System Strengths**

* **Offline Functionality:** No internet required, making the system highly portable and private.
* **Real-Time Feedback:** Users received guidance within ~4 seconds, which felt immediate and helpful.
* **Custom Object Classes:** The system was able to detect user-defined objects not included in COCO, such as bags or specific types of bottles.
* **Hands-Free Interaction:** Complete control using speech, freeing the user's hands and attention.

**6.5 Custom YOLOv11n Model Results:**

* Precision: 90%
* Recall: 86%
* F1 Score: 85%
* mAP@0.5: 89.3%

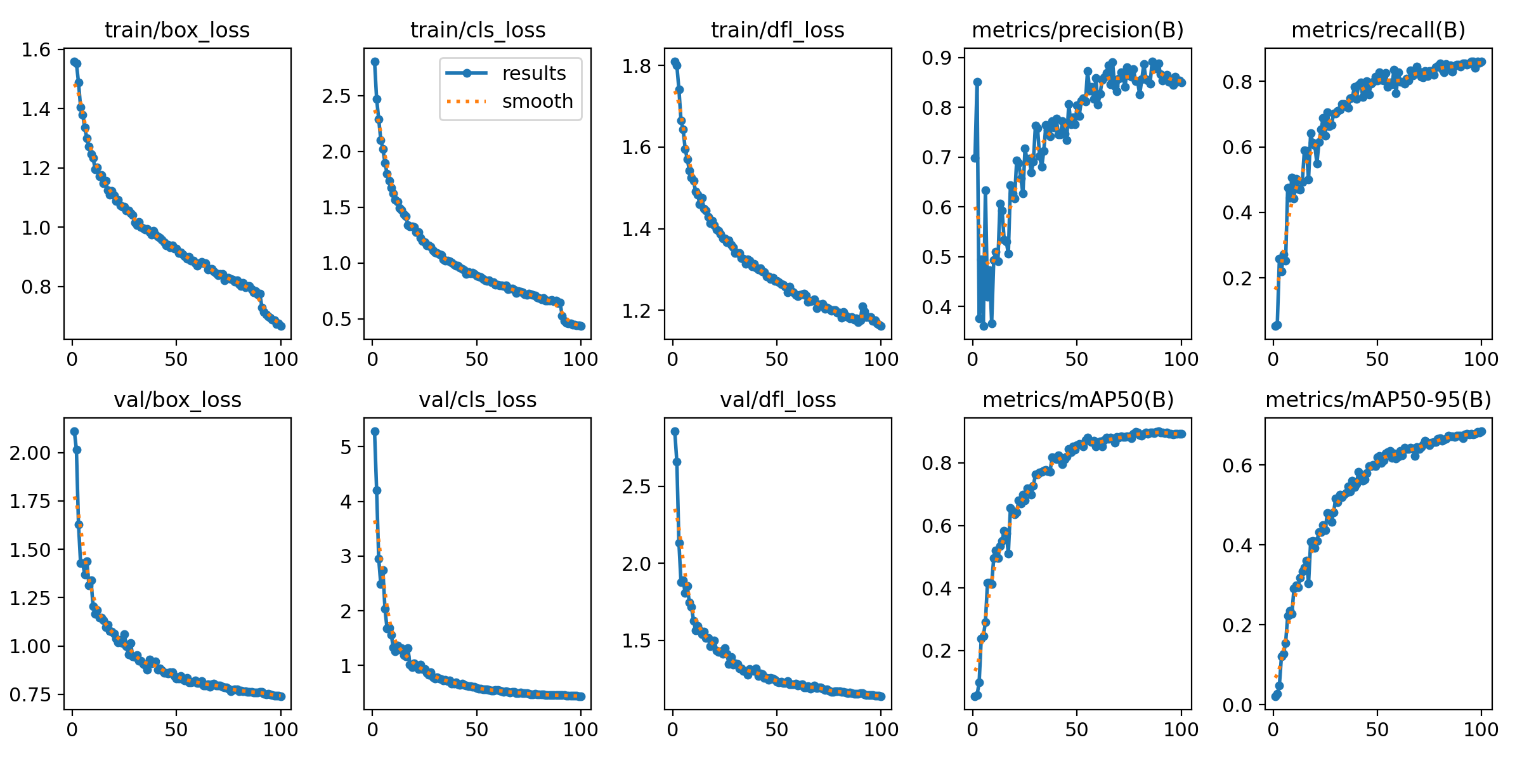
This chart shows the training progress of the YOLOv11 model. All losses go down, while precision, recall, and mAP scores go up. This means the model is learning well and performing better over time.

Figure 11 : YOLOv11 training and validation curves

This is the normalized confusion matrix for the custom YOLOv11 model. It shows high accuracy for most classes like **Door (96%)**, **Fire Extinguisher (94%)**, and **Light Switch (100%)**. Some confusion appears with classes like **Water Cooler** and **Trash Can**, and the **background class** has some overlap with multiple categories.

A screenshot of a graph

AI-generated content may be incorrect.

Figure 12 : Normalized confusion matrix

These results indicate that the model performs well in recognizing and localizing commonly encountered indoor objects with high confidence.

**6.6 Summary Of Discussion**

The project met its main objectives, both technically and in terms of usability. SightOfBlind demonstrated that an embedded, real-time assistive system using modern AI techniques could offer genuine value to visually impaired users — not just in theory, but in practice.

These findings will serve as the foundation for future iterations of the system with improved accuracy, robustness, and environmental awareness.

# Chapter 7: Limitations

Although the SightOfBlind system successfully achieves its core objectives, several limitations were identified during development and testing. These reflect technical constraints, usability boundaries, and environmental factors, providing a roadmap for future improvements.

**1. Sensitivity to Lighting Conditions**

The RGB webcam used by the system requires good lighting for accurate object detection. In dark environments or under harsh lighting, detection confidence drops. The system does not yet support low-light enhancement, night vision, or infrared input.

**2. Environmental Noise and STT Challenges**

Speech-to-text (STT) performance suffers in noisy or echo-prone settings. The AirPods microphone offers clarity, but in crowded or open spaces, background noise often interferes with recognition accuracy. The system lacks noise filtering or adaptive gain control.

**3. Fixed Object Class Limitation**

The YOLOv11n model was trained on a limited set of indoor objects. It cannot detect objects outside this list unless retrained. This restricts adaptability to changing environments or personalized needs without developer intervention.

**4. Raspberry Pi Processing Limits**

Even with the Raspberry Pi 5’s improved hardware, resource constraints limit the use of higher-resolution models or additional features (like person tracking or obstacle avoidance) without affecting frame rate. The system had to be carefully optimized to balance speed and detection accuracy.

**5. No Path Planning or Obstacle Avoidance**

SightOfBlind helps locate objects but does not guide users around obstacles or suggest navigation paths. Without ultrasonic or LiDAR input, it cannot detect or respond to floor hazards, walls, or moving obstacles between the user and the object.

**6. Simplified Direction Logic**

Object direction is calculated based on static horizontal zones (left, center, right). This lacks depth estimation, vertical positioning, or rotational orientation — making feedback occasionally vague in cluttered or complex scenes.

**7. Heat and Processing Load**

The Raspberry Pi 5 runs the object detection model continuously, causing heat buildup over time. While a cooling fan is included, prolonged use in enclosed or warm areas may still lead to thermal throttling or reduced performance.

**8. High Voltage Power Requirement**

The Raspberry Pi 5 requires a stable 5V/5A power supply, especially when running a webcam, Bluetooth, and real-time inference. This exceeds the output of many standard power banks, making power management a challenge in mobile, wearable settings. Users must carry a high-capacity, high-output power source, which affects portability and runtime.

**9. Lack of Real-World Testing with Blind Users**

The system has not yet been tested by visually impaired individuals in daily life. Current testing has been simulated using blindfolded users. Real feedback from the intended user base is essential to validate comfort, usability, and trust.

Despite these limitations, SightOfBlind remains a functional and innovative prototype that lays a strong foundation for real-world assistive solutions. Each identified constraint also opens a path for future technical and user-experience enhancements.

# Chapter 8: Conclusion

# The *SightOfBlind* project presents a practical, AI-powered solution to enhance the independence and spatial awareness of visually impaired individuals through real-time object detection and voice interaction. By combining state-of-the-art computer vision (YOLOv11n), natural voice processing, and embedded hardware (Raspberry Pi 5), the system enables users to locate and identify objects in indoor environments using simple voice commands — without the need for internet connectivity or external assistance.

# Throughout the development process, a custom dataset was collected and annotated to fine-tune the detection model for indoor objects relevant to blind users. The integration of speech recognition and text-to-speech technologies allowed the system to interact naturally with users in a hands-free, intuitive manner. Extensive testing demonstrated that the system operates efficiently in real time, providing accurate guidance and responsive feedback in various indoor settings.

# While the system performed well overall, several limitations were identified, including sensitivity to lighting and noise, fixed object class support, and power consumption challenges. Despite these constraints, the prototype successfully achieved its core objectives and proved that wearable, offline assistive technology can be made affordable, accurate, and practical.

# The project successfully fulfilled the specific objectives outlined in Chapter 3. A lightweight and wearable hardware setup was designed using a Raspberry Pi 5, USB webcam, AirPods microphone, cooling case, and portable power bank, allowing the system to function reliably in indoor environments. Real-time object detection was achieved using a YOLOv11n model, incorporating both the original and a custom-trained version focused on indoor objects. Voice interaction was implemented through speech-to-text processing using the AirPods microphone, enabling users to issue natural commands such as “find the chair.” These commands were parsed to identify target objects and trigger the detection module. Detected objects were analyzed for direction (left, center, right), and clear voice feedback was delivered through gTTS over Bluetooth. The system was thoroughly tested in various indoor scenarios, where it demonstrated accurate object recognition, responsive voice interaction, and practical usability—meeting all core project goals.

# This project not only demonstrates the feasibility of real-time assistive AI on embedded devices but also lays the groundwork for future development in intelligent mobility aids. It reflects the potential of accessible technology to empower visually impaired individuals with greater confidence and autonomy in their everyday lives.

# Chapter 9: Future Work

**Future Work**

While the SightOfBlind system has proven to be an effective and usable prototype, there are several areas where the system can be improved, expanded, and adapted to provide even greater value to visually impaired users. The following enhancements are proposed for future development:

**1. Expand Object Detection Classes**

Future versions of the system should support a broader range of object categories by:

* Collecting a larger, more diverse dataset.
* Incorporating user-customized object detection.
* Allowing the model to learn new objects on-device with simplified fine-tuning tools.

**2. Add Obstacle Detection and Navigation**

Integrating additional sensors such as ultrasonic sensors or depth cameras could enable:

* Real-time obstacle avoidance,
* Path planning,
* Safer navigation in unfamiliar or dynamic environments.

**3. Improve Directional Feedback**

Instead of simple left-center-right logic, future versions could:

* Include distance estimation using stereo vision or LiDAR,
* Offer more descriptive feedback (e.g., “The door is 2 meters to your left”),
* Use compass/GPS orientation to describe directions (e.g., “northwest corner”).

**4. Enhance Voice Interaction**

Adding natural language processing (NLP) support could make voice interaction more flexible and conversational, enabling commands like:

* “Is there a bottle near the door?”
* “Tell me what’s around me.”
* “What objects do you see?”

**5. Multimodal Feedback (Vibration/Haptics)**

To improve accessibility for users in noisy environments or with hearing impairments, future versions could:

* Use vibration motors or haptic actuators,
* Support customizable vibration patterns for directional cues.

**6. Reduce Power and Heat Requirements**

Future hardware optimization may involve:

* Switching to lower-power AI boards (e.g., Coral Edge TPU, NVIDIA Jetson Nano),
* Improving thermal management,
* Extending battery life through software optimization.

**Vision Forward**

With ongoing research in embedded AI, wearable computing, and human-AI interaction, SightOfBlind has strong potential to evolve from a prototype into a real-world assistive product. Its future lies in being smarter, more adaptive, and more inclusive — ultimately helping visually impaired individuals experience their surroundings with greater confidence, clarity, and freedom.

# References

1. Hersh, M., & Johnson, M. A. (2008). *Assistive Technology for Visually Impaired and Blind People*. Springer.  
   <https://link.springer.com/book/10.1007/978-1-84628-867-8>
2. Bourbakis, N. (2008). *Artificial Vision for the Blind via a Wearable Vision System*. IEEE Engineering in Medicine and Biology Magazine.  
   <https://ieeexplore.ieee.org/document/4479601>
3. Microsoft. *Seeing AI – Talking Camera App for the Blind*.  
   <https://www.microsoft.com/en-us/ai/seeing-ai>
4. Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. arXiv preprint.  
   <https://arxiv.org/abs/1804.02767>
5. Ultralytics. *YOLOv11 Model Documentation*.  
   <https://docs.ultralytics.com/models/yolo11>
6. Python Software Foundation. *gTTS: Google Text-to-Speech*.  
   <https://pypi.org/project/gTTS/>
7. OrCam Technologies. *OrCam MyEye – Wearable AI Device for the Blind*.  
   <https://www.orcam.com>
8. Itoh, Y., et al. (2021). *AI Suitcase: A Smart Travel Assistant for the Visually Impaired*.  
   <https://dl.acm.org/doi/10.1145/3411764.3445143>
9. Roboflow. *Annotation and Data Preparation Platform*.  
   <https://roboflow.com>
10. *SpeechRecognition Python Library*.  
    <https://pypi.org/project/SpeechRecognition/>