TEAM - D14

AI-POWERED SMART GOGGLES FOR VISUALLY IMPAIRED PEOPLE

INTEGRATED WITH DAA AND BIOLOGY

MHOKESH
CB.SC.U4AIE23316

KOWSHIK
CB.SC.U4AIE23314

VIBHU SANCHANA
CB.SC.U4AIE23347

ASMI K
CB.SC.U4AIE23351

OBJECTIVE

The objective of this project is to develop **AI-powered smart goggles** that assist visually impaired individuals in **safe and independent navigation**. The system integrates **Robotics and Mathematical optimization to provide real-time guidance** without internet dependency.

For obstacle detection, the goggles **use YOLOv11 for object recognition** and ultrasonic and infrared sensors to identify objects, **even in low-light conditions**. For navigation, a GPS module with the **A algorithm* calculates the best route to the user's destination.**

Users can select their destination **using voice commands**, making the system fully hands-free. **Real-time feedback** is provided through audio instructions via a speaker and haptic feedback through vibration motors when obstacles are too close.

This project ensures **fully offline functionality**, allowing visually impaired individuals to navigate confidently in various environments without requiring internet access.

LITERATURE REVIEW COMPARISON

Paper	Drawback/Challenge	Our Methodology to Overcome
Design and Construction of Electronic Aid for Visually Impaired People (Patil et al)	Limited obstacle detection system with basic sensors; lacked adaptability to complex environments.	Incorporated YOLO v11 for accurate obstacle detection, ultrasonic sensors for depth detection, and interactive voice feedback for user assistance.
Unifying terrain awareness through real-time semantic segmentation (Yang et al)	Depth camera-based system was costly and unsuitable for outdoor environments.	Used cost-effective Raspberry Pi 4, Pi Camera, and ultrasonic sensors for indoor and outdoor navigation with reliable obstacle detection.
wearable face recognition system on google glass for assisting social interactions (Mandal et al.)	Relied solely on Google Glass for face recognition, lacking additional navigational support.	Introduced vibratory motors and voice commands to provide multi-modal feedback, enhancing situational awareness for visually impaired users.

LITERATURE REVIEW COMPARISON

Paper	Drawback/Challenge	Our Methodology to Overcome
Advanced Indoor and Outdoor Navigation System for Blind People Using Raspberry Pi. (Anandan et al)	Focused only on obstacle detection without navigation or night-time usability.	Added IR sensors for reliable night-time navigation and a 10,000mAh battery for extended operation in real-world environments.
Smart guiding glasses for visually impaired people in indoor environment (Bai et al.)	System required depth cameras, making it expensive and limited in practical use.	Leveraged affordable ultrasonic sensors for depth detection and integrated a lightweight system using Raspberry Pi and Pi Camera.
A novel approach to wearable image recognition systems to aid visually impaired people (Chen et al)	Heavily dependent on pre-trained datasets and lacked adaptability to dynamic environments.	Used YOLO v11 pre-trained on general datasets and Raspberry Pi 4 for real-time adaptability to various obstacles and environmental conditions.
Envision	Required internet connectivity for core functionalities, limiting offline usability.	Designed the system to work entirely offline, ensuring reliability in areas without internet access.

LITERATURE REVIEW COMPARISON

Paper	Drawback/Challenge	Our Methodology to Overcome
. Implementation of Multi-Object Recognition System for the Blind (Park et al.)	Lacked night-time capabilities and was limited to object recognition in daylight conditions.	Integrated IR sensors to enable night-time usability, ensuring consistent functionality across different lighting conditions.
Oxsight	Did not provide depth detection or tactile feedback for obstacle awareness.	Added ultrasonic sensors for precise depth detection and vibratory motors to deliver immediate tactile feedback for nearby obstacles.
Deep learning based face recognition system with smart glasses. (Daescu et al.)	Relied on 5G connectivity for system operation, limiting its functionality in offline scenarios.	Implemented a completely offline system powered by a 10,000mAh battery to ensure continuous operation in remote areas.
Pedestrian detection with wearable cameras for the blind (Lee et al.)	Focused solely on face recognition without navigation or night-time capabilities.	Expanded functionality to include obstacle detection, depth sensing, and IR-based night-time navigation with interactive voice commands.

RESEARCH GAP

1 **User-Specific Adaptability & Customization**

- Current alerts are predefined (audio/vibration), limiting user preference.

🔹 **Future Work:** Add customizable alert settings (e.g., tone-based alerts, adjustable vibration intensity).

2 **Night-Time Performance Optimization**

- IR sensors help, but low-light object recognition is still a challenge.

🔹 **Future Work:** Implement thermal cameras or night-vision algorithms for better detection.

3 **Real-Time Processing Efficiency**

- Raspberry Pi 4 may face latency issues when running YOLO v11 + voice feedback together.

🔹 **Future Work:** Explore hardware accelerators like Google Coral TPU or NVIDIA Jetson Nano.

4 **Wearability and Ergonomics**

- 10,000mAh battery is currently pocket-mounted, which may be inconvenient.

🔹 **Future Work**: Optimize battery placement for lightweight wearability.

5 Lack of Multi-Language Support

- Voice commands work in one language (English), limiting usability for non-English speakers.

🔹 **Future Work:** Add multilingual speech recognition & text-to-speech (TTS).

METHODOLOGY

Object Detection and Recognition (Using AI and Image Processing) :

The camera continuously captures images and processes them using YOLOv11, a deep-learning model specialized in object detection.

How YOLOv11 Detects Objects :

- **Image Capture:** The camera takes real-time images of the surroundings.
- **Feature Extraction:** YOLOv11 processes the image, identifying key features of objects like people, vehicles, or obstacles.
- **Bounding Box Prediction:** The model draws boxes around detected objects, indicating their location.
- **Confidence Scoring**: It assigns a confidence score to each detected object, ensuring only the most relevant detections are used.
- **Filtering Duplicates:** The system removes overlapping or incorrect detections to ensure accurate object recognition.

To make object detection more efficient, **Singular Value Decomposition (SVD)** is used to refine image data.

SVD helps in **Removing noise, Highlighting important features, Improving speed and accuracy.**

METHODOLOGY

GPS-Based Navigation (Finding the Best Route) :

The system guides users to their destination using GPS data and intelligent pathfinding algorithms.

Tracking Location Using GPS :

The **Neo-6M GPS module** continuously updates the user’s location in terms of **latitude and longitude**. This helps determine **how far the user is from their destination** and whether they are moving in the **right direction.**

Finding the Shortest Path Using A* :

Once the user provides a voice command for their destination, the system calculates the best path using the **A* (A-star) algorithm**. This method:

- Analyzes all possible paths and **selects the shortest and safest route**.
- Accounts for obstacles and **avoids areas that might be unsafe.**
- **Recalculates the path dynamically** if the user moves in the wrong direction.

The system continuously monitors the user's progress, adjusting directions in real-time to ensure smooth navigation.

METHODOLOGY

Obstacle Detection Using Sensors :

The smart goggles use **ultrasonic and infrared sensors to detect obstacles** in the user's path.

Detecting Obstacles with Ultrasonic Sensors :

Ultrasonic sensors send sound waves and measure how long it takes for the sound to bounce back. This helps determine the distance to obstacles, such as walls, poles, or moving objects.

Enhancing Obstacle Detection in Low-Light Conditions :

Infrared (IR) sensors are used alongside ultrasonic sensors to detect obstacles in dim light or in environments where the camera might not work well. IR sensors measure reflected infrared light to identify objects and their proximity.

Combining Sensor Data for Better Accuracy :

Since both ultrasonic and IR sensors provide obstacle information, the system merges data from both sensors to improve accuracy.

This makes sure that the system accurately detects obstacles and provides timely alerts.

METHODOLOGY

Feedback System (Audio and Vibration Alerts) :

To help users understand their surroundings, the system provides real-time feedback through **voice instructions and vibrations.**

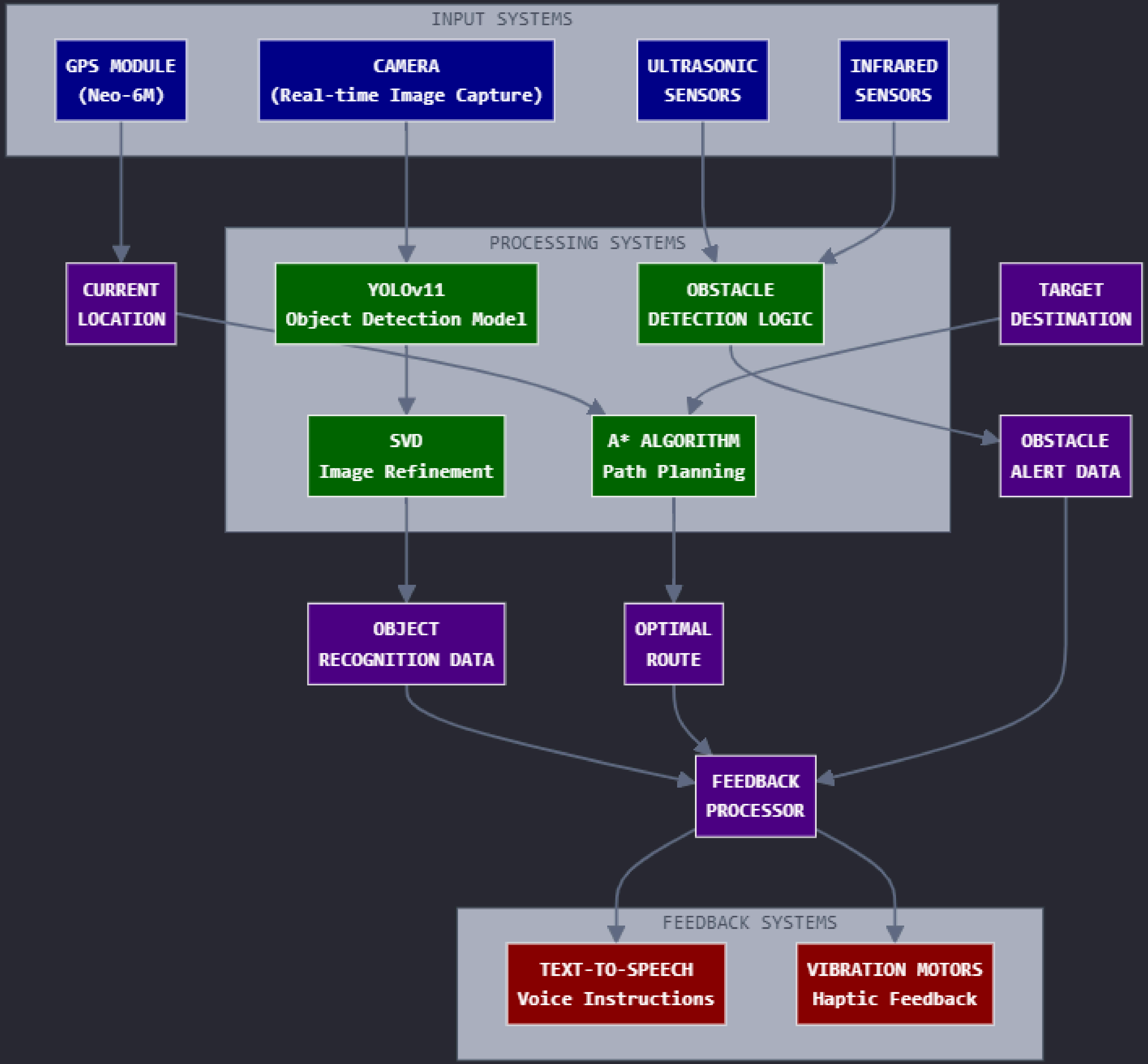
Voice Instructions for Navigation :

- A **Text-to-Speech (TTS) module** converts system-generated instructions into spoken commands.
- The user hears **voice guidance** about the nearest objects, the direction they should move, and potential hazards.

Vibration-Based Obstacle Alerts :

- Vibration motors provide **haptic feedback** when an obstacle is detected.
- The intensity of vibrations increases as the user gets closer to an obstacle.

By combining audio and vibration feedback, the system ensures that users receive timely alerts even in noisy environments where audio feedback might not be effective.


INPUT SYSTEMS
GPS MODULE (Neo-6M)
CAMERA (Real-time Image Capture)
ULTRASONIC SENSORS
INFRARED SENSORS
PROCESSING SYSTEMS
CURRENT LOCATION
YOLOv11 Object Detection Model
OBSTACLE DETECTION LOGIC
TARGET DESTINATION
SVD Image Refinement
A* ALGORITHM Path Planning
OBSTACLE ALERT DATA
OBJECT RECOGNITION DATA
OPTIMAL ROUTE
FEEDBACK PROCESSOR
FEEDBACK SYSTEMS
TEXT-TO-SPEECH Voice Instructions
VIBRATION MOTORS Haptic Feedback

MATHEMATICAL CONCEPTS

1. Object Detection with YOLOv11 :

The smart goggles use YOLOv11 to detect obstacles and guide the user safely. The detection process involves:

- **Bounding Box Prediction:** The model identifies objects by drawing rectangular boxes around them.
- **Confidence Scores:** The probability that an object belongs to a certain class.
- **Intersection over Union (IoU):** A metric to measure the accuracy of detection.

Bounding Box Calculation :

- **x,y** → Center coordinates of the bounding box
- **w,h** → Width and height of the bounding box

These values help in precisely **locating obstacles** in the camera's view.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Intersection over Union (IoU) for Object Accuracy :

To check if the **detection is correct**, the system compares the predicted bounding box with the actual object location. This is done using IoU:

- If **IoU > 0.5** , the detection is considered accurate.
- If multiple boxes detect the same object, **Non-Maximum Suppression (NMS)** removes less confident detections, keeping only the most relevant one.

MATHEMATICAL CONCEPTS

2. Singular Value Decomposition (SVD) for Feature Optimization :

- Object detection (filtering noise in image features).
- Sensor fusion (combining ultrasonic and IR sensor data).

SVD in Image Processing :

YOLOv11 represents images as feature matrices. SVD breaks down these matrices into simpler parts:

$$A = U\Sigma V^T$$

Where:

- A = Original image feature matrix
- U = Left singular vectors (object structure)
- Σ = Singular values (feature importance)
- V^T = Right singular vectors

SVD in Sensor Fusion :

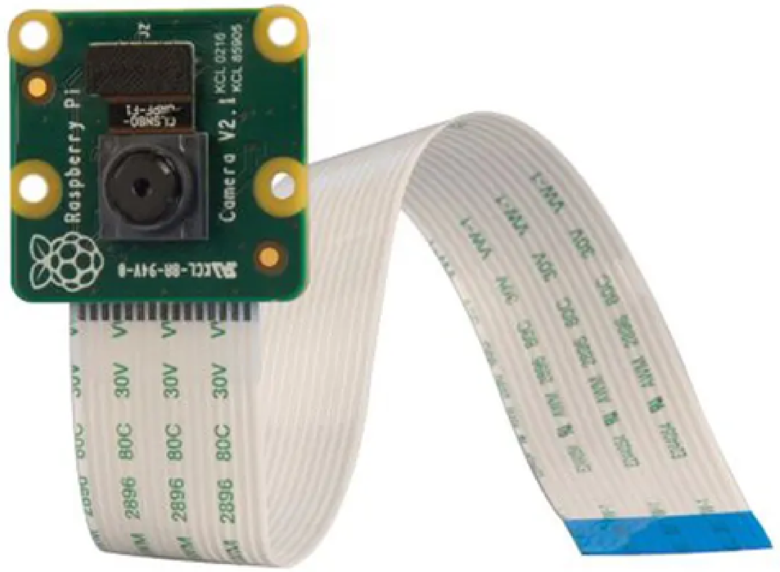
- To combine ultrasonic and IR sensor data, **SVD removes redundant information and highlights important signals.**
- If one sensor gives inaccurate data, **SVD filters it out**, improving obstacle detection accuracy.

Example: If ultrasonic data is unstable, SVD uses IR sensor values to provide a more reliable distance reading.

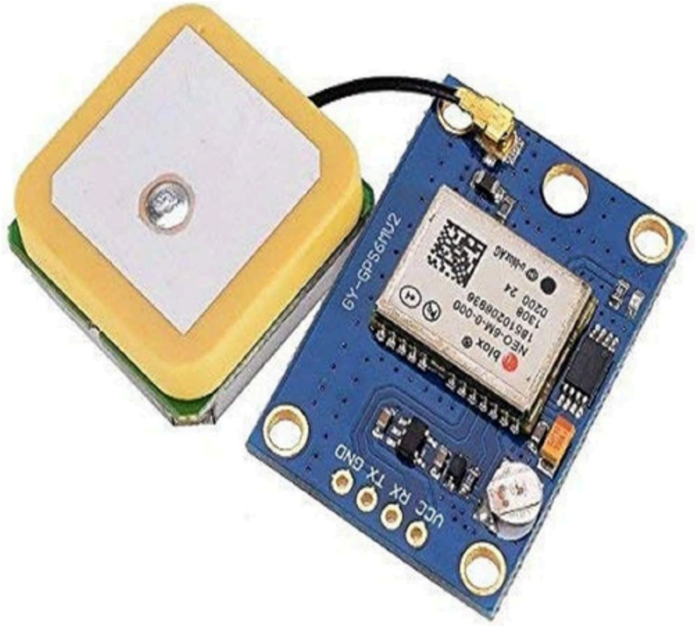
HARDWARE

- **Processing Unit:** Raspberry Pi – Runs AI models and processes sensor data.
- **Camera Module:** Raspberry Pi Camera – Captures real-time images for object detection.
- **GPS Module:** Neo-6M GPS – Provides real-time location tracking.
- **Ultrasonic Sensors:** HC-SR04 – Detects nearby obstacles using sound waves.
- **Infrared (IR) Sensors:** Used to detect obstacles in low-light environments.
- **Vibration Motors:** Provide haptic feedback when an obstacle is close.
- **Headphones:** Deliver voice-based navigation and obstacle alerts.
- **Microphone:** Captures user voice commands for navigation.
- **Battery Pack:** Provides portable power for continuous usage.
- **SD Card:** At least 32GB for storing the operating system and AI models.

Raspberry Pi

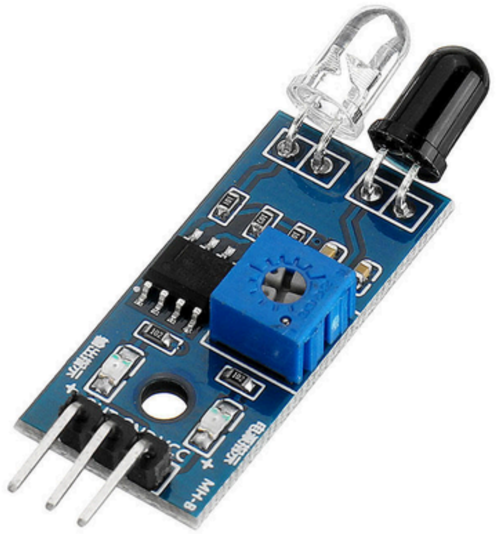


Raspberry Pi Camera



Neo-6M GPS

HC-SR04



Infrared sensors

Vibration motors

Headphone and Microphone

SD Card

SOFTWARE

- **Operating System:** Raspberry Pi OS.
- **Programming Language:** Python – Main language for AI model integration and sensor communication.
- **Computer Vision Library:** OpenCV – Used for image processing and real-time video analysis.
- **AI Model:** YOLOv11 – Performs object detection and classification.
- **Machine Learning Framework:** PyTorch or TensorFlow-Runs the deep learning model.
- **GPS Data Processing:** Geopy – Used to calculate distance and navigation paths.
- **Pathfinding Algorithm:** A* Algorithm – Finds the best route for navigation.
- **Text-to-Speech (TTS) Engine:** pyttsx3 – Converts text alerts into voice instructions.
- **Sensor Communication Libraries:**

    RPi.GPIO – Controls ultrasonic and IR sensors.

    smbus – Handles I2C communication for external modules.

TIMELINE

1 Research & Planning

Research existing solutions and technologies (e.g.,Raspberry Pi, OpenCV). Decide on hardware components (camera, speaker, microcontroller,etc). Choose programming languages and frameworks (Python, C++,TensorFlow, etc.). Identify speech synthesis and object detection techniques.

2 Hardware Assembly & Testing

Procure and assemble hardware components. Integrate the camera, speaker, and microcontroller. Test individual components for functionality (camera clarity, speaker output, etc.).

3 Software Development

Implement object detection using OpenCV/TensorFlow. Develop text-to-speech (TTS) conversion for audio feedback. Integrate real-time voice output based on detected objects. Test and debug software functionalities.

4 Integration & Optimization

Merge hardware and software for seamless operation. Optimize power consumption and latency. Enhance accuracy of object detection and voice output. Conduct field tests with real users for feedback.

5 Final Testing & Documentation

Perform extensive testing in different environments. Fix any remaining bugs or issues.

THANK YOU