S.A.G.E - Personalized AI-Powered Smartglass

Ananya P (22CSB08 MDL22CS036)
M Gayathri (22CSB36 MDL22CS126)
Navneet Ranjish Pillai (22CSB41 MDL22CS138)
Nikhil M (22CSB44 MDL22CS144)

Model Engineering College

October 2025

Contents

1. Introduction
2. Why S.A.G.E?
3. Features
4. Problem Statement
5. Literature Survey
6. Proposed Solution
7. Software Requirements Specification (SRS)
8. Functional Requirements
9. Non-Functional Requirements

Contents

10 Software Design Document (SDD)

11 System Architecture

12 Use Case Diagram

13 Data Flow Diagram

14 Module Description

15 Conclusion

16 References

Introduction

Introduction

- S.A.G.E is designed to enhance real-world interaction through vision and voice-based features
- Built around a Raspberry Pi Zero 2 W, with a focus on software-driven functionality and minimal hardware complexity
- Integrates a custom Flutter mobile app to manage AI tasks, user controls, and real-time communication
- Features voice-triggered assistant, HUD-based feedback, facial recognition, object detection, and translation
- **S.A.G.E** : Situational Awareness and Guidance Engine. It's a situational companion designed to empower users with seamless, context-aware insights, wherever they go

Why S.A.G.E?

Why S.A.G.E?

- Traditional assistive and wearable tech often lacks intelligence, real-time adaptability, and affordability.
- S.A.G.E leverages AI to provide intuitive, voice-based interaction and real-time visual understanding.
- Enhances accessibility, especially for visually impaired users, by offering on-demand object and face recognition.
- Offloads complex tasks like translation and environment understanding to a companion app and hosted AI backend.
- Bridges the gap between affordability, functionality, and intelligent personal assistance in a compact form factor.

Why S.A.G.E?

- The system integrates specialized AI modules to handle distinct smartglass functionalities:
  - Voice-activated assistant for hands-free interaction,
  - Real-time object and facial recognition for environmental awareness,
  - Text translation through live camera input,
  - HUD display rendering for visual feedback.
  - Navigation assistance
- Enables on-demand AI features, triggered by voice, to assist users contextually and adaptively.
- SAGE redefines affordable, intelligent wearables by merging accessibility, AI capabilities, and intuitive UX in a compact form.

Features

HUD Interaction & Visual Recognition

Heads-Up Display (HUD)

- **Live Visual Feedback:** Reflects real-time information such as navigation or object labels.
- **Context-Aware Display:** Dynamically updates based on voice commands and environmental input.

Visual Recognition

- **Object Detection:** Identifies objects in the user's environment using ML models.
- **Facial Recognition:** Detects and recognizes familiar faces.

System Architecture & Module Collaboration

Modular Design

- **Distributed Responsibilities:** Divide tasks across Raspberry Pi, mobile app, and hosted backend (e.g., UI, ML inference, internet APIs).
- **Optimized Processing:** Run lightweight tasks on-device and delegate intensive operations to the app/server.

Module Collaboration

- **Voice, Vision, and Display Integration:** Modules communicate via local server hosted by Raspberry Pi.
- **Synchronized Workflow:** Seamless coordination enables real-time feedback through HUD and speaker.

Customization & Performance Assurance

User-Centric Customization

- **Context-Based Interaction:** Customize behavior based on user needs - accessibility, productivity, or travel.
- **Modular Feature Control:** Enable or disable object scanning, translation, or music control as needed.

Performance Assurance

- **Latency Optimization:** Offload AI tasks to backend while maintaining responsive HUD and voice processing.
- **Thermal Efficiency:** Offloaded ML models avoid overheating and maintains usability.

Problem Statement

Problem Statement

Existing smart wearable technologies are often expensive, rely on proprietary platforms, and overlook real-world accessibility needs. Visually impaired users face challenges in navigating and interacting with their surroundings, creating the need for an affordable, hands-free solution that enhances real-time environmental awareness. It aims to develop a low-cost, open-source smartglass that prioritizes accessibility, usability, and seamless hands-free interaction.

Literature Survey

BASE PAPER - Smart Navigation Glasses For Independent Living[1]

Sathya, V., Preetha, H., Gopika, K.

- A smart wearable system combining ESP32-CAM, ultrasonic sensors, and Raspberry Pi for real-time obstacle detection
- Integrates OpenCV and TensorFlow Lite for machine learning-based object and facial recognition
- Modules include text-to-speech, obstacle discovery, facial recognition, and SOS messaging
- Provides both online and offline object detection with Clarifai and TensorFlow for robust performance
- Includes Bluetooth connectivity, step tracking, ambient temperature sensing, and FTP for media transfer
- Designed for accessibility, affordability, and adaptability to aid visually impaired individuals in daily navigation

Visual Information Translator Using Smart Glasses for Blind[3]

Rani, T.P., Vignesh T., Susila Sakthy S., Priyadharshan M., Kalaichelvi P.

- IoT and Machine Learning-based smart glasses system for the visually impaired
- Equipped with Pi Camera, Microcontroller, Bluetooth/WiFi module, and mobile app interface
- Modules include text-to-speech conversion, object and obstacle detection, and facial recognition
- Uses Google ML Kit and TensorFlow Lite for real-time image processing and feedback
- Features SOS alert system, health report database, and interactive voice feedback to improve autonomy and safety

Smart Glasses Embedded with Facial Recognition Technique[7]

Kumar, M., Bharti, B., Chauhan, U.

- Raspberry Pi-based smart glasses system for visually impaired, with facial recognition and obstacle detection
- Uses ESP32 camera, ultrasonic sensors, Google Text-to-Speech, and Python for real-time assistance
- Identifies known individuals and objects from a preloaded database, announcing identity and distance
- Ultrasonic sensor alerts user about unrecognized obstacles or persons, enhancing spatial awareness
- Designed to be cost-effective, lightweight, and easily extendable with additional modules for various assistive functions

Facial Recognition Smart Glasses for Visually Impaired People[6]

Ghodake, A., Sale, H., Kamble, A., Mankari, K., Lad, O., Mishra, V.

- Smart glasses system equipped with ESP32-CAM and Raspberry Pi for facial recognition and obstacle detection
- Uses OpenCV and Haar Cascade classifiers to identify known individuals and objects
- Implements text-to-speech module via Google API to convey visual information through audio
- Designed to recognize multiple faces and alert users via audio output with high accuracy and reliability
- Aims to reduce dependence on external assistance, promoting safety and independence for visually impaired users

Single-Shot Image Recognition Using Siamese Neural Networks[5]

Malhotra, A.

- Introduces a Siamese Neural Network model for one-shot image recognition tasks
- Designed to compare a query image against a reference image set and determine similarity
- Employs contrastive loss to learn embeddings that minimize distance between similar pairs and maximize between dissimilar pairs
- Suitable for applications where limited training data is available and conventional classification fails
- Demonstrated effectiveness in facial recognition, signature verification, and handwriting recognition tasks

Smart Glasses for Visually Impaired People using Machine Learning[2]

Siva Priyanka S., Aruru Sai Kumar, M V Nagabhushanam, Dasari Vennela, Pallakila Divya Tulasi

- Proposes smart glasses using object detection (YOLOv3) and text-to-speech (Google API) for assisting the visually impaired.
- Raspberry Pi 3 used as the core processor, with a Pi camera module and speaker for feedback.
- OpenCV is used for real-time object detection; Google TTS API converts detected objects into audio prompts.
- Enhances independent mobility and awareness by alerting users of surrounding objects.
- Achieved high object recognition accuracy, e.g., 99

Smart Glass for Visually Impaired Person[4]

Sabitha R., Senthil Pandi S., Rathi Devi J., Jegan G.

- IoT-enabled smart glasses integrating GPS, ultrasonic sensors, ESP32-CAM, and Bluetooth for real-time navigation and emergency support
- Provides obstacle detection with audio feedback and alerts emergency contacts with live video and location in unfamiliar environments
- Includes mobile app for managing settings, monitoring, and communication with emergency contacts
- Reduces notification fatigue by distinguishing between familiar and unfamiliar locations
- Utilizes YOLO for object recognition and GPS for geo-fencing, enhancing autonomous mobility for visually impaired users

Indoor Navigation Glasses for the Visually Impaired[8]

Pocholo James Loresco, Rence Jerome C. Cruz, Kingsley Z. Ramones, Julia Angellica D. Zafra, Karl Russell G. Ramirez

- Proposes wearable smart glasses for indoor navigation using deep learning and audio guidance for visually impaired individuals.
- Utilizes Raspberry Pi 4B, Coral Edge TPU, and RPi Camera v2 housed in 3D-printed glasses for processing and vision.
- Incorporates MobileNetV2 for room localization and MobileNet-SSD for obstacle detection (e.g., chairs).
- Voice-activated via Vosk and PyAudio; synthesizes directional cues like "go left" or "obstacle in front."
- Achieved high performance: Navigation accuracy of 98.96

Google Pi Using Raspberry Pi for Visually Impaired[4]

Katkar, G. V., Abirami, K., Posonia, A. M., Ankayarkanni, B., Ushanandini, D.

- Introduces Google Pi – a smart assistant for the visually impaired using Raspberry Pi, Google APIs, and deep learning.
- Integrates real-time voice commands, object detection (YOLOv3), and currency recognition (CNN).
- Achieves high accuracy: 92.4
- Enhances accessibility with custom voice commands, portability, and real-time feedback via audio.
- Demonstrates significant improvement over traditional voice assistants lacking object and currency identification features.

Literature Survey (1/3)

Title	Authors	Key Findings	Advantages	Disadvantages	Inferences
Smart Glasses for Visually Impaired People using Machine Learning	Siva Priyanka S, Aruru Sai Kumar, M V Nagabhushanam, Dasari Vennela, Pallakila Divya Tulasi	• Uses Raspberry Pi 3 with camera module YOLOv3 for object detection • OpenCV and Google API for TTS provide real-time obstacle detection and audio alerts.	• Real-time detection • speech guidance • low cost • OpenCV support.	• Limited detection classes may struggle in cluttered scenes.	• Boosts mobility and safety using YOLO on embedded hardware.
Visual Info Translator Glasses Using Smart Glasses for Blind	T.P. Rani, Vignesh T, S Susila ,P Kalaichelvi	• Uses OCR and text-to-speech to convert printed text into audio output for blind users via smart glasses.	• Real-time visual-to-audio conversion • portable and non-intrusive design.	• Limited to printed text • not effective for complex visual scenes or handwriting.	• Effective for reading printed material • extends accessibility to printed media for the visually impaired.
Google Pi using Raspberry Pi for Visually Impaired	Gowthami Vinod Katkar, K Abirami, A. Mary Posonia, B. Ankayarkanni, D. Ushanandini	• Introduces "Google Pi" — a smart assistant built on Raspberry Pi with Google Assistant API, YOLOv3, and CNN for real-time object. • Currency recognition to support visually impaired people. Features voice control, object identification, and financial assistance.	• Combines Google Assistant with deep learning for real-time feedback, user-friendly voice control, and portable. • Affordable hardware, and independence for visually impaired.	• Requires significant training data and computing power for CNN models. • No facial recognition or OCR included yet. • Performance may vary with lighting and background.	• Demonstrates the power of integrating Raspberry Pi with Google APIs. • Demonstrates CNN models to deliver assistive technology that enhances autonomy and accessibility for the visually impaired.

Literature Survey (2/3)

Title	Authors	Key Findings	Advantages	Disadvantages	Inferences
Smart Glass for Visually Impaired Person	Sabitha R, Senthil Pandi S, Rathi Devi J, Jegan G	• GPS, ultrasonic sensors, emergency camera, and Bluetooth-enabled mobile app • alerts sent during unfamiliar environment detection.	• Emergency alerts • recognizes safe zones • video and GPS feedback.	• Bluetooth limitations • phone dependency.	• Offers both navigation and emergency support through IoT.
Single-Shot Image Recognition Using Siamese Neural Networks	Abhiraj Malhotra	• Siamese Neural Network used for one-shot learning image recognition • high accuracy with low data.	• Few-shot learning, efficient with limited data, low memory use.	• Poor generalization on unseen classes, sensitive to noise.	• Suitable for applications with minimal samples • potential in face or object recognition.
Facial Recognition Smart Glasses for Visually Impaired People	Asha Ghodake, Hanamant Sale, Abhijeet Kamble, Krushna Mankari, Om Lad, Vinit Mishra	• Raspberry Pi based system with ultrasonic sensor and webcam • integrates Pytesseract for OCR and TTS for audio output. Facial recognition + obstacle distance estimation.	• Headphones for feedback, multiple detection supported.	• Limited face DB accuracy; audio delay possible.	• Simplifies obstacle and face recognition using Raspberry Pi.

Literature Survey (3/3)

Title	Authors	Key Findings	Advantages	Disadvantages	Inferences
Smart Glasses Embedded with Facial Recognition Technique	Mahim Kumar, Bhawana Bharti, Usha Chauhan	• Uses Raspberry Pi 4, Google TTS, ultrasonic sensors, facial recognition, and ESP32 camera • adds web browsing, music.	• Multifunctional: voice assistant + face/object detection.	• Power supply needs, bulkiness due to multiple components.	• Combines assistive and entertainment features with accessibility.
Indoor Navigation Glasses for the Visually Impaired with Deep Learning and Audio Guidance Using Google Coral Edge TPU	Pocholo J. Loresco, Rence J. C. Cruz, Kingsley Z. Ramones, Julia A. D. Zafra, Karl R. G. Ramirez	• Uses MobileNet-SSD + Google Coral Edge TPU for real-time navigation, localization, obstacle avoidance • audio cues via AUI • camera detects chairs for indoor path guidance.	• High accuracy indoor navigation, low latency, no need for external beacons.	• Only trained on specific objects (e.g., chair), requires Coral Edge TPU.	• Highly effective indoor assistive device using deep learning and audio UI.

Proposed Solution

Proposed Solution

S.A.G.E is a wearable device that integrates machine learning, IoT, and human interaction to provide a hands-free, intelligent user experience. The system is built around a **central processor** and a **mobile application** for user interaction and cloud connectivity.

- **AI Assistant**
- **Heads-Up Display (HUD)**
- **Machine Learning**
- **Database-Powered System**
- **Client-Server Architecture**

Software Requirements Specification (SRS)

Product Perspective

- Integrates wearable hardware (S.A.G.E) with a companion mobile app.
- Utilizes AI services for real-time natural language and vision processing.
- Replaces traditional glass functions with intelligent features: HUD, voice commands, translation.
- Modular design facilitates easy software updates and hardware upgrades.
- Supports seamless communication over Wi-Fi hotspot created by the mobile device.

Product Functions

- Voice-activated wake word detection and command recognition.
- Real-time heads-up display for recognition results.
- Object and face recognition via camera feeds and ML models.
- Support for real-time multi-language translation of text and speech.
- Mobile app interface for user interaction and AI backend communication.
- Secure pairing and fallback network recovery modes.

User Classes and Characteristics

- End Users: Require hands-free assistance, including visually impaired and tech enthusiasts.
- Developers: Implement hardware and software, integrate APIs, and maintain the system.

Operating Environment

- Python-hosted server on Raspberry Pi.
- Companion mobile application on Android.
- Wi-Fi hotspot network established by mobile device for internet access.
- Cloud AI APIs accessible via Internet through mobile app.
- Hardware peripherals: camera, microphone, TFT display, buttons.

Design and Implementation Constraints

- Limited processing power and thermal budget on wearable hardware.
- Dependence on mobile device hotspot means network quality varies.
- Battery life constraints require efficient power management.
- API dependencies require stable external AI service availability.
- User interface must remain simple to accommodate diverse users.

User Interfaces

- Mobile app supports voice commands, feature toggling, and status monitoring.
- S.A.G.E HUD displays navigation, translation, and recognition results.
- Audio feedback complements visual cues to aid accessibility.

Hardware Interfaces

- Raspberry Pi, camera, display, and buttons.
- USB microphones ensure clear voice input.
- Speakers provide necessary audio output.
- Device powered using consistent power draw from power banks.

Software Interfaces

- REST APIs facilitate real-time data exchange between S.A.G.E and mobile app.
- AI integrations include Google Vision, Gemini, and LibreTranslate APIs.
- HTTPS ensures encrypted and secure communications.
- Modular design allows easy addition of new AI services.
- On-device software handles voice activation and HUD display rendering.

Communications Interfaces

- Primary communication over Wi-Fi hotspot created by paired mobile device.
- Use of TCP/IP with HTTPS for secure and reliable data transmission.
- Bluetooth Low Energy used for pairing and device discovery.
- Communication protocols optimized for low latency and bandwidth efficiency.

Hardware Requirements

- Wearable computing platform: Raspberry Pi
- High-resolution camera module for image capture and streaming.
- TFT heads-up display (HUD) for overlay visuals.
- Microphone array with noise-cancelled voice input via USB.
- Speakers for discreet audio output.
- Wireless modules supporting Wi-Fi and BLE for device pairing.

Software Requirements

- Python-hosted server on-device.
- Modules for image capture, streaming, and communication with mobile app.
- Companion mobile app using **Flutter** for Android managing AI services.
- Integration with AI(Gemini, Google Vision, LibreTranslate).
- Secure HTTPS for data transmission.
- On-device HUD rendering and audio feedback system.
- Firmware for microphone, cameras and power management.

Functional Requirements

1. Wake Word Detection and Voice Commands

- System continuously listens for "Hey Sage" to start the device and ready-up for voice commands.
- Captures and processes commands for navigation, translation, object recognition, etc.
- High-priority feature for user interaction.

2. Heads-Up Display (HUD) Features

- Displays navigation cues, translations, and recognition results in real-time.
- Supports dynamic content updates with minimal latency.

3. Image Capture and Streaming

- Captures images every second via voice command.
- Streams securely to the mobile app or stores locally for batch processing.

4. AI-Based Recognition and Translation

- Performs object detection and recognition using AI models.
- Supports facial recognition.
- Provides real-time text and speech translation between languages.

5. Mobile Application Integration

- Mobile app manages API calls, preferences, and updates.
- Ensures secure communication and synchronization with S.A.G.E.

6. Device Pairing and Network Management

- Supports setup via access point mode for pairing.
- Handles credentials and network configuration securely.

Non-Functional Requirements

Performance Requirements

- Latency below 2 seconds for voice/AI responses.
- Safe thermal operation during extended use.
- Battery life of 3 - 5 hours active usage.

Safety Requirements

- Compliance with wearability and electronic safety standards.
- Prevent overheating via thermal management.

Security Requirements

- Secure authentication in the mobile app.
- Encrypted network communications.
- Secure storage of credentials.
- Privacy-aware handling of facial and image data.

Software Quality Attributes

- Modularity for easy maintenance.
- Scalable for adding new AI features.
- Reliable with graceful error recovery.
- Simple and user-friendly interfaces.

Software Design Document (SDD)

System Architecture

System Architecture

The system features modular architecture with components handling:

- Object detection
- Facial recognition
- Language translation
- Navigation
- Voice assistance

All modules communicate for synchronized operation.

System Architecture

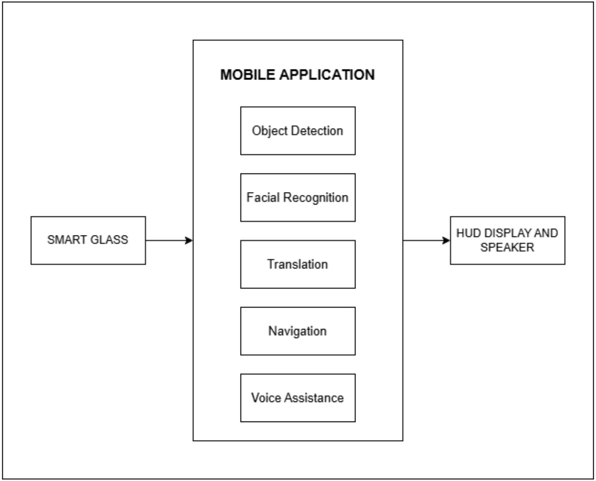


Figure: System Architecture Diagram

Use Case Diagram

Use Case Diagram

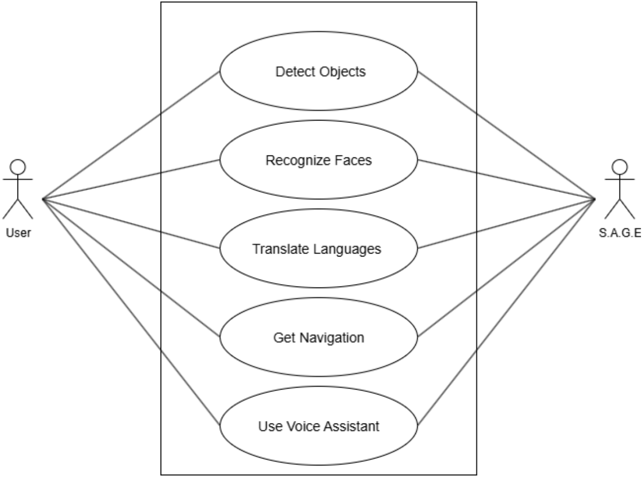


Figure: Use-case diagram

Data Flow Diagram

Data Flow Diagram

LEVEL 0:

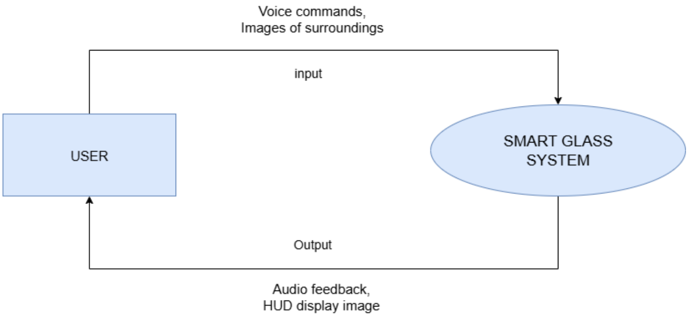


Figure: Level 0 Data Flow Diagram

Data Flow Diagram

LEVEL 1:

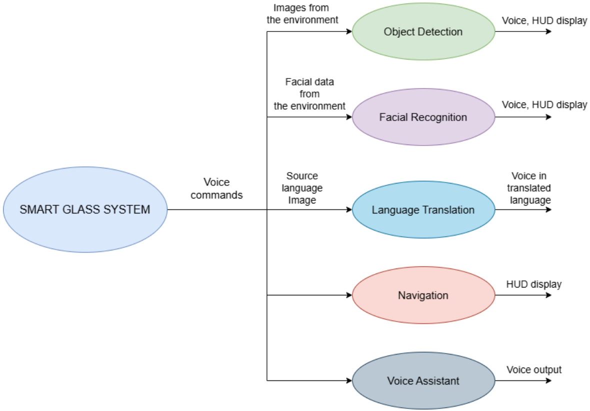
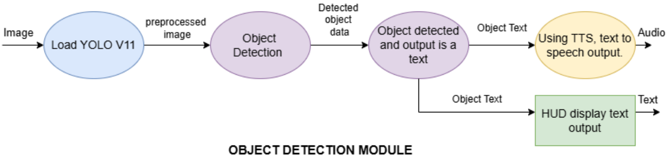


Figure: Level 1 Data Flow Diagram

Data Flow Diagram

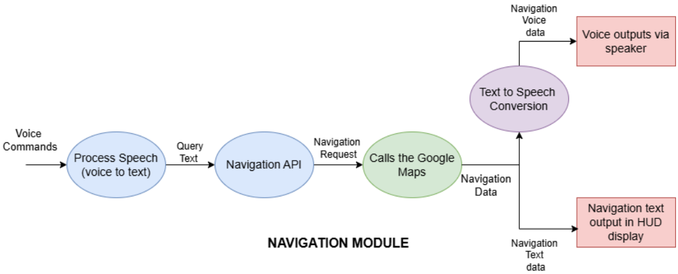
LEVEL 2.1:



OBJECT DETECTION MODULE

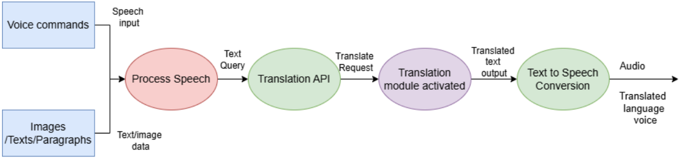
Data Flow Diagram

LEVEL 2.2:



Data Flow Diagram

LEVEL 2.3:



LANGUAGE TRANSLATION MODULE

Data Flow Diagram

LEVEL 2.4:

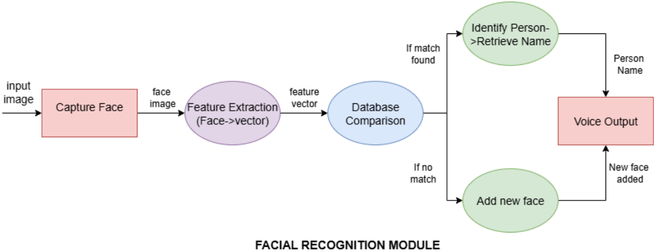


Figure: Level 2.4 Data Flow Diagram

Data Flow Diagram

LEVEL 2.5:

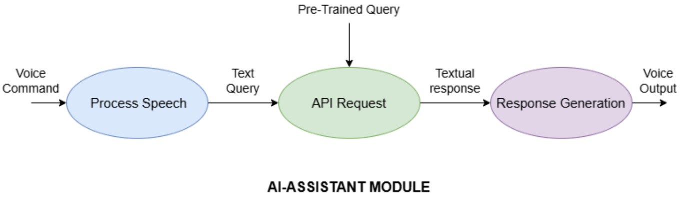


Figure: Level 2.5 Data Flow Diagram

Module Description

- **Object Detection Module**
- **Facial Recognition Module**
- **Language Translation Module**
- **Navigation Module**
- **Voice Assistant Module**

Main Algorithm - S.A.G.E (Part 1)

Algorithm System Workflow

Input: User voice commands captured through the S.A.G.E microphone

Start

Initialize the Voice Assistant module.

while S.A.G.E is active **do**

- Continuously listen for the wake phrase **"Hey S.A.G.E"**.
- Once activated, capture and process the user's voice command.
- Analyze the intent of the user's command.
- **if** user requests **Object Detection then**
  - Switch control to the **Object Detection Module**.

Main Algorithm - S.A.G.E (Part 2)

- **else if** user requests **Facial Recognition then**
  - Switch control to the **Facial Recognition Module**.
- **else if** user requests **Language Translation then**
  - Switch control to the **Language Translation Module**.
- **else if** user requests **Navigation then**
  - Switch control to the **Navigation Module**.
- **else**
  - Continue conversation or execute general tasks using the **Voice Assistant Module**.
- Return to listening mode for the wake command.

End while

End

Object Detection Module

- Captured frames are transmitted to the connected mobile application for processing.
- Utilizes the YOLO deep learning model for efficient and accurate detection.
- Displays identified objects through the S.A.G.E HUD and provides audio feedback.
- Helps users recognize surrounding objects and improves environmental awareness.

Algorithm

Algorithm Object Detection Module

Input: Real-time visual feed from the camera.

Start

while camera captures frames **do**

- Transmit image to the connected mobile application.
- Load YOLOV11 for real-time object detection.
- YOLOV11 returns Pre-processed image which is of upscaled quality.
- Object detection is activated.
- **if** objects are detected **then**
  - Display object label on the HUD.
  - Convert detection results to audio feedback via TTS.

End

Facial Recognition Module

- This module identifies and recognizes individuals in real-time.
- Captures facial images through the camera.
- Converts faces into **feature vectors** and compares them with stored entries.
- Announces recognized individuals' names via **audio output**.
- Stores unrecognized faces in the database for future identification.

Algorithm

Algorithm Facial Recognition Module

Input: Live facial images captured through the camera

Start

while camera captures a face **do**

- Convert captured face to a feature vector.
- Compare vector with stored database entries.
- **if** match found **then**
  - Announce person's name and description.
- **else**
  - Store new face as a new entry.
- Output as voice.

End

Language Translation Module

- Enables translation of text or visual content from the environment.
- Detects and extracts text using Google Vision from the camera feed.
- Sends extracted text to the **LibreTranslate API** for language translation.
- Converts translated text into voice output for user convenience.
- Aids users in understanding signboards, instructions, and written materials.

Algorithm

Algorithm Language Translation Module

Input: Visual or text-based content captured from the environment

Start

while text is detected **do**

- **if** voice input **then**
  - Apply Speech-To-Text processing on the voice input.
- **else if** image input **then**
  - Capture image snapshot.
  - Send the Captured image to the mobile application.
  - Process the image to extract textual data using Google Vision.
- Send text to LibreTranslate API for translation.
- Receive translated output (e.g., French to English).
- Deliver translation through voice output on speaker.

Navigation Module

- Provides real-time route guidance and navigation to the user.
- Integrates with the **Google Maps API** for accurate route generation.
- Displays live directions on HUD.
- Offers **hands-free voice guidance** for safe navigation.

Algorithm

Algorithm Navigation Module

Input: GPS data and map requests from mobile application

Start

while navigation is active **do**

- Apply Speech-To-Text processing on voice input.
- Pre-process the query with the received command.
- Initialize the Navigation by prompting a Google Maps API request.
- Display route details on HUD, and output through speakers for acessibility.

End

Voice Assistant Module

- Enables voice-based interaction between the user and S.A.G.E.
- Activates using a wake phrase such as "**Hey S.A.G.E**".
- Processes voice commands using the **Gemini API**.
- Provides conversational responses and executes control actions.
- Allows hands-free operation and enhances accessibility for users.

Algorithm

Algorithm Voice Assistant Module

Input: Voice commands captured through the microphone

Start

while voice assistant is active **do**

- **if** wake phrase "**Hey S.A.G.E**" is detected **then**
  - Process user command using **Gemini API**.
  - Provide conversational voice responses or control actions.

End

Database Design

Database uses normalized relational structure:

- **Users:** Authentication details and preferences.
- **Detection Logs:** Object and facial detection records.
- **Navigation History:** Routes and travel logs.

Foreign keys maintain data integrity across entities.

Conclusion

- **S.A.G.E** is a wearable device that integrates machine learning, IoT, and human interaction to provide a hands-free, intelligent user experience.
- Provides real-time object detection, facial recognition, translation, and voice interaction to enhance accessibility.
- Modular design ensures scalability and adaptability for future improvements.
- Demonstrates the potential of AI-driven wearables in human assistance.
- Future work: gesture control, emotion recognition, and better battery optimization.

References (1/3)

- V. Sathya; Preetha. H; Gopika. K, in *Smart Navigation Glasses For Independent Living*", IEEE, 2023, doi:10.1109/ICCCT63501.2025.11020288. Available: https://ieeexplore.ieee.org/document/11020288
- S.PriyankaS., A.S. Kumar, M.V. Nagabhushanam, D.Vennela, and P.D. Tulasi, in *"Smart Glasses for Visually Impaired People using Machine Learning*", IEEE, 2023, pp.1–6. doi:10.1109/ICCCNT56998.2023.10307374. Available: https://ieeexplore.ieee.org/document/10307374
- T.P. Rani, V.T. Vignesh, S.Susila, and P.Kalaichelvi, in *"Visual Information Translator Using Smart Glasses for Blind*", IEEE, 2023, doi:10.1109/ICCEBS58601.2023.10449230 Available: https://ieeexplore.ieee.org/document/10449230
- G.Vinod Katkar, K.Abirami, A.Mary Posonia, B.Ankayarkanni, and D.Ushanandini, in *"Google Pi using Raspberry Pi for Visually Impaired*", IEEE, 2024, doi: 10.1109/AIIoT58432.2024.10574549 Available: https://ieeexplore.ieee.org/document/10574549

References (2/3)

[4] R.Sabitha, S.PandiS., J.Rathi Devi, and G.Jegan, in *"Smart Glass for Visually Impaired Person"* , IEEE, 2023, doi:10.1109/ICCCT63501.2025.11019089 Available: https://ieeexplore.ieee.org/document/11019089

[5] A.Malhotra, in *"Single-Shot Image Recognition Using Siamese Neural Networks"*, IEEE, 2023, doi:10.1109/ICACITE57410.2023.10182466 Available: https://ieeexplore.ieee.org/document/10182466

[6] A.Ghodake, H.Salé, A.Kamble, K.Mankari, O.Lad, and V.Mishra, in *"Facial Recognition Smart Glasses for Visually Impaired People"*, doi: 10.1109/ICCUBEA58933.2023.10391977 Available: https://ieeexplore.ieee.org/abstract/document/10391977

References (3/3)

- M.Kumar, B.Bharti, and U.Chauhan, in *"Smart Glasses Embedded with Facial Recognition Technique"*, IEEE, 2023, doi:10.1109/AISC56616.2023.10085337 Available: `https://ieeexplore.ieee.org/document/10085337`
- P.J. Loresco, R.J.C. Cruz, K.Z. Ramones, J.A.D. Zafra, and K.R.G. Ramirez, in *"Indoor Navigation Glasses for the Visually Impaired with Deep Learning and Audio Guidance Using Google Coral Edge TPU"*, IEEE, 2024, doi:10.1109/TENCON61640.2024.10902929 Available: `https://doi.org/10.1109/TENCON61640.2024.10902929`

Thank You