Graduation Project

Intelligent Mixed Reality Navigation Assistant for Smart Buildings with Multimodal AI Integration

Submitted By
Team

1. **Ahmed Mohamed Mousaa - 222101392**
2. **Sandy Samy Samir – 222101524**
3. **Aya Wael Elshamy – 22210383**
4. **Mariam Khaled Hassan - 222101358**
5. **Alaa Adel Elsayed – 222101567**
6. **Aya tarek mahmoud – 222101469**
7. **Basma Ahmed Mahmoud - 221101164**

Project Advisor
<Title> <Name> <LastName>

Faculty of Computer Science and Engineering
New Mansoura University

2025-2026

Graduation Project

SMART GLASSES

Submitted By
Team

Student Name	Student Academic ID	Program	Track
Ahmed Mohamed Mousaa	222101392	AIS	
Sandy Samy Samir	222101524	AIS	
Aya Wael Elshamy	22210383	AIS	
Mariam Khaled Hassan	222101358	CS	
Alaa Adel Elsayed	222101567	CS	
Aya tarek mahmoud	222101469	CS	
Basma Ahmed Mahmoud	221101164	AIE	

ABSTRACT

This project presents the design and development of an advanced smart glasses system aimed at enhancing daily communication, productivity, and human–technology interaction. The system integrates multiple cutting-edge technologies including speech recognition, real-time translation, large language models, computer vision, augmented reality, GPS navigation, environmental sensing, and smart home connectivity. The glasses provide a seamless and hands-free user experience, allowing individuals to communicate effortlessly across language barriers, record audio and video, capture photos, take notes, browse the web, and access information through a built-in heads-up display (HUD).

A YOLO11-based computer vision module is trained on custom datasets to detect objects and recognize faces, enabling the system to identify members of the user's social circle and personalize interactions. Navigation is supported through a custom indoor GPS model built using manually mapped building structures, graph-based routing, and Wi-Fi triangulation to provide accurate guidance even inside complex environments. The speech recognition pipeline uses Whisper to transcribe user speech in Arabic or English and forward it to a large language model, which interprets tasks and outputs structured JSON instructions. A multilingual text-to-speech engine then generates natural responses in the user's preferred language.

The hardware is built around an ESP32 module for Bluetooth and Wi-Fi communication, connected to microphones, speakers, sensors, and the smart glasses interface. A companion mobile and web dashboard enables smart home control, device management, and analytics to provide broader accessibility and convenience. The system is also designed with inclusivity in mind, offering voice commands, haptic feedback, visual aids, and personalized assistance for users with disabilities or mobility challenges. It further benefits drivers by reducing interaction friction and improving navigation safety.

Overall, the project delivers a comprehensive, modular, and scalable smart glasses platform that improves daily communication, enhances decision-making, and increases productivity through intelligent, natural, and context-aware technology. The integration of AI, AR, and multimodal interaction positions this system as a next-generation wearable solution suitable for education, healthcare, business, travel, customer service, and beyond. (Will be added here.)

Additionally, the project incorporates a comprehensive competitor analysis covering Amazon Echo Frames and Meta Ray-Ban glasses, highlighting the technological gaps our system solves through advanced computer vision, indoor GPS, smart home integration, and AI-driven personalization. The architecture follows a modular and scalable design, supported by detailed diagrams, structured user flows, and a clearly defined feature list reflecting the full ecosystem of the smart glasses platform. The system includes a responsive mobile app and a web dashboard that complement the glasses by offering device control, analytics, and accessibility settings.

ACKNOWLEDGEMENTS

(Optional section) Thank to the people who extensively contributed to your study.

TABLE OF CONTENTS

LIST OF TABLES

Category	Apple Vision Pro	Oppo Air Glass 3	Xiaomi Wireless AR	Huawei Vision Glass	Meta Quest 3
Device Type	MR Headset	AR Glasses	AR Glasses (wireless)	Smart Display Glasses	Mixed Reality Headset
Price	$3,499+	~Estimated $800	—	~$430	$499 (128GB)
Release Year	2024–2025	2024	2023 Prototype	2023	2023
Weight	750–800 g + 353 g battery	~50 g	40–120 g	Lightweight	~515 g
OS / Platform	visionOS	ColorOS (phone)	Xiaomi AR OS	N/A	Meta OS (Android-based)
CPU / Chipset	Apple M5 + R1	Phone-based	Snapdragon XR2	Phone-based	Snapdragon XR2 Gen 2
GPU	10-core GPU	Phone GPU	Adreno	Phone GPU	Adreno 740
RAM	16GB unified	—	—	—	—
Storage	256–1TB	—	—	—	128–512GB
Display Type	Micro-OLED	Waveguide	Micro-OLED waveguide	Micro-OLED	LCD + Pancake lenses
Resolution (per eye)	23M pixels total	—	High PPD	1080p	2064×2208
Refresh Rate	90/96/100/120 Hz	—	—	N/A	120 Hz
Field of View	—	—	—	N/A	—
Brightness	—	1000+ nits	~1200 nits	High	—
Sensors	14+ sensors: LiDAR, IMUs, eye tracking	Touch, mic	3 cams + IMU	Basic	RGB/IR cams, depth
Cameras	Stereo 3D + passthrough	—	Tracking cams	—	2 RGB + 4 IR
Tracking	Hand + eye + head	Basic	Hand + head	None	Inside-out 6DoF
Hand Tracking	Yes	Yes	Yes	No	Yes
Eye Tracking	Yes	No	No	No	No
Authentication	Optic-ID (iris)	—	—	—	—

Audio	Spatial Audio	On-ear speakers	Spatial audio	Stereo	3D spatial
Battery	2.5 h (external)	Unknown	Concept	None	2–3 h
Charging	USB-C	—	—	USB-C	USB-C 3.2
Connectivity	Wi-Fi 6, BT 5.3	BT, Wi-Fi	Wireless low-latency	USB-C	Wi-Fi 6E, BT
Input Methods	Eye/hand/voice	Touch	Hand/voice	Buttons	Controllers + hand
IPD Range	51–75 mm	—	—	—	—
Use Cases	Full MR apps	Smart overlay	AR apps	Virtual screen	VR/MR gaming
Special Features	Spatial video, OpticID	High portability	Wireless AR	Cinema display	Full-color passthrough

Category	**Magic Leap 2**	**HoloLens 2**	**Meta Ray-Ban Glasses**	**Lenovo ThinkReality A3**	**VIVE XR Elite**	**Amazon Echo Frames (3rd Gen)**	**Smart Glasses**	Amazon Amelia Smart Glasses (Best-Estimate)
Device Type	Enterprise AR Headset	Enterprise MR Headset	Smart Glasses	AR Tethered Glasses	XR Headset	Smart audio glasses (no AR display)	Smart glasses (AR + sensor-rich wearable)	Enterprise AR smart glasses for delivery workers
Price	$3,600–$5,500	~$3,500	$299–$379	$1,499	$1,099	269$~	Estimated $200 (without EoS)	Not announced (expected enterprise-rate ~$1,200–$1,800)
Release Year	2022	2019	2023	2021	2023	3) 2023rd Gen(	2026	2025
Weight	260 g (headset)	566 g	48–50 g	130 g	Modular, lightweight	36–31~g depending on lens type	–70 120g	110–70g EST. (glasses only; battery offloaded to vest)
OS /	Magic	Windo	Meta	Custom	Androi	Amazon	Linux	Custom

Platform	Leap OS	ws Holographic	firmware	Android	d XR	custom firmware	(Raspberry Pi OS / other Linux distro)	Amazon OS (Android-based, estimated)
CPU / Chipset	AMD Zen 2 + RDNA2	Snapdragon 850 + HPU 2.0	Qualcomm mobile SoC	PC/Phone CPU	Snapdragon XR2	Low-power embedded audio chipset	Processor Server-based (PC offload)	Snapdragon XR1 / XR2-class (estimated)
GPU	RDNA2	Adreno + HPU	Integrated	PC/phone GPU	Adreno 650	None	Server-based (PC GPU)	Integrated Adreno GPU EST
RAM	16GB	4GB	—	PC RAM	12GB	Not specified (low-power embedded RAM)	8GB onboard + PC RAM	4–2GB EST.
Storage	256GB NVMe	64GB	—	PC storage	128GB	Not applicable	128 GB onboard + PC storage	64–32 GB onboard EST.
Display Type	Waveguide	Holographic waveguide	None	Micro-OLED	LCD + pancake	)no AR / no HUD(	TBD	Monocular or binocular micro-LED / micro-OLED HUD EST
Resolution (per eye)	1440×1760	2048×1080	—	1920×1080	1920×1920	—	TBD	640×400 – 1280×720
Refresh Rate	120 Hz	60 Hz	—	60 Hz	90 Hz	—		
Field of View	70°	52°	—	~40°	110°	—	TBD	°20–10 EST
Brightne	20–2000	~500	—	200–400	—	—	TBD	+1000

ss	nits	nits		nits				nits (for outdoor delivery use) EST
Sensors	3 cams + depth + eye	Depth + IR eye tracking	IMU, touch, mics	IMU, tracking cams	Depth + RGB + IMU	Accelero meter	IMU-Eye trackin g-Stereo 3D-Touch-Micro phone-Camer a-Speake r-Blueto oth radio	IMU (accelero meter + gyro)
Cameras	12.6MP RGB + depth	RGB + depth	12MP	8MP RGB	16MP RGB	None		based CV hazard detection
Tracking	SLAM + eye tracking	6DoF + hand + eye	None	6DoF	Inside-out 6DoF	Basic motion tracking (IMU only)	TBD	Type No full 6DoF; uses basic IMU + visual tracking for alignmen t EST.
Hand Tracking	Yes	Yes	No	Yes	Yes	No	Yes	None
Eye Tracking	Yes	Yes	No	No	No	No	Yes	None
Authenti cation	Iris ID	—	LED indicat or only	—	—	—		
Audio	Spatial	Spatial	Open-ear speake rs	Stereo	Integra ted speaker s	Open-ear directiona l speakers (4-micro	Unkno wn	Small open-ear speakers or bone-

						speaker array)		conduction EST
Battery	3.5 h	2–3 h	3–4 h	Tethered	~2 h (swap)	6~hours continuous playback / full day mixed use	Life Unknown	Life 8–12 hours using vest battery pack
Charging	USB-C	Fast charge	Case	USB-C	USB-C	2~hours	USB-C	Time 1–2 hours for vest battery EST
Connectivity	Wi-Fi 6, BT	Wi-Fi 5	Bluetooth	USB-C tether	Wi-Fi 6E, BT	Proprietary charging cable	Wi-Fi, Bluetooth, Wired (PC link)	Bluetooth + Wi-Fi (tethered to vest/phone/hub)
Input Methods	Hand/eye/voice	Hand/eye	Voice/touch	Controller optional	Controllers	Voice (Alexa)	Eye, Hand, Voice, Touch, Buttons	Vest-mounted controller + Voice commands
IPD Range	—	Auto	—	—	54–73 mm	—	TBD	N/A (fixed HUD, no IPD adjustment)
Use Cases	Enterprise AR	Enterprise MR	Camera + audio	Enterprise workflows	XR gaming + MR		Camera + Audio + Controller + AR	Delivery routing
Special Features	Dynamic dimming, SLAM	Holo UI	Social capture	PC-class AR	Hot-swap battery	Auto-off when removed	Hot-swap battery	Hazard detection, Privacy mode

LIST OF FIGURES

SYMBOLS & ABBREVIATIONS

ACM: Association for Computing Machinery

APA: American Psychological Association

IEEE: Institute of Electrical and Electronics Engineers

1. INTRODUCTION

1.1. Problem Statement

In today's fast-paced world, technology has become an integral part of daily life. However, users face several challenges when interacting with various technological devices, particularly in daily communication, productivity, and navigation. Key problems include:

1. Language Barriers: Users often struggle to communicate effectively due to differences in spoken or written language, especially in international environments or while traveling.

2. Task Distraction: Managing multiple tasks at once—such as note-taking, capturing photos, recording audio, or checking messages—can overwhelm users and reduce efficiency.

3. Lack of Intelligent Assistance: Most devices do not provide a personal smart companion capable of understanding user needs and offering real-time guidance.

4. Indoor Navigation Limitations: In large or complex buildings like universities, hospitals, or corporate campuses, finding the desired destination can be difficult without accurate navigation support.

5. Accessibility Constraints: Current technology often lacks integrated solutions to support users with disabilities, limiting their ability to interact fully with devices.

The proposed system, Smart Glasses, addresses these challenges by combining AI-driven smart assistance, speech recognition, real-time translation, augmented reality, indoor navigation, and environmental and health monitoring sensors into a single portable and ergonomic device.

1.2. Project Purpose

The primary goal of the Smart Glasses project is to create a productive, intelligent, and user-friendly environment that enhances daily life by:

- Enabling seamless hands-free communication through speech recognition and translation features.

- Providing a smart companion capable of understanding tasks, managing schedules, setting reminders, and offering personalized recommendations.

- Improving navigation in complex indoor and outdoor environments using GPS, computer vision, and graph-based pathfinding algorithms.

- Supporting users with disabilities through voice commands, visual aids, haptic feedback, and gesture recognition.

- Allowing immersive experiences with augmented reality overlays for both productivity and entertainment purposes.

- Continuously adapting to user behavior using machine learning algorithms for personalized interaction and decision support.

The project aims to reduce daily friction, enhance productivity, and improve the overall quality of life, providing a comprehensive solution that integrates communication, navigation, health monitoring, and intelligent assistance in one device.

1.3. Project Scope

The scope of the Smart Glasses project includes:

1. Hardware Development:

- Wearable device platform with microphone, speakers, camera, display, and sensors.

- GPS module for location-based services and indoor positioning.

- Low-latency wireless connectivity (Wi-Fi, Bluetooth).

2. Software Development:

- Speech recognition and text-to-speech (TTS) systems.

- Natural Language Processing (NLP) for task understanding.

- AI-powered smart companion for scheduling, reminders, and assistance.

- Machine learning models for adaptive and personalized user experience.

- Augmented reality interface for immersive visualization and interaction.

- Web browsing and dashboard interface for remote management.

3. User Interaction Features:

- Hands-free operation with voice commands and gesture recognition.
- Real-time translation for multiple languages.
- Environmental and health monitoring with alerts and recommendations.
- Seamless communication with loved ones.

The project is designed to cater to a wide range of use cases including education, healthcare, business, entertainment, travel, customer service, and accessibility for users with disabilities.

1.4. Objectives and Success Criteria of the Project

The objectives of the Smart Glasses project are:

a. Enhance Communication: Provide real-time transcription and translation to overcome language barriers.

b. Improve Daily Productivity: Allow multitasking such as note-taking, video recording, and quick information access.

c. Provide Smart Assistance: Develop a companion AI that understands user commands and context.

d. Enable Accurate Navigation: Integrate GPS and indoor positioning for precise route guidance.

e. Ensure Accessibility: Support users with various disabilities with inclusive features.

f. Offer Immersive AR Experiences: Overlay relevant information on the real world for improved situational awareness.

g. Continuously Adapt: Implement machine learning models to learn from user behavior and preferences.

Success Criteria:

- Smooth operation of all hardware and software components.
- Accurate speech recognition and translation in multiple languages.
- Effective navigation and indoor positioning performance.

- Positive user feedback on usability, comfort, and accessibility.
- Continuous adaptation and improvement of user experience over time.

1.5. Report Outline

1. Introduction

This chapter introduces the problem addressed by the project, the purpose and scope of the work, and the criteria used to measure its success. It also outlines the structure of the report.

2. Related Work

This chapter reviews existing smart glasses technologies, indoor navigation systems, voice interaction interfaces, smart home automation solutions, and AI-based wearable devices. It summarizes the limitations of current systems and compares them with the proposed solution.

3. Methodology

This chapter explains the steps taken during the project, including requirement analysis, system design, implementation methods, tools and technologies, and testing strategies. It also provides an overview of the dataset, the models used (YOLO11, Whisper, LLM, NLP pipeline), and the proposed integrated approach.

4. Experimental Results

This chapter presents the results of the implemented system, including object detection accuracy, indoor navigation path precision, speech recognition performance, smart home interactions, and hardware functional tests.

5. Discussion

This chapter interprets the results, discusses system performance, evaluates limitations, and analyzes challenges encountered during implementation such as hardware constraints, GPS accuracy indoors, and LLM processing performance.

6. Conclusions

This chapter summarizes the achievements of the project, the value it provides for users (including drivers and people of determination), and recommendations for future enhancements.

References

This section lists all sources, research papers, datasets, tools, and libraries used in the project following the required academic citation format.

Appendix

The appendix contains supplementary materials such as code snippets, hardware specifications, additional diagrams, JSON path maps, complete AR/VR interface screenshots, and raw data samples.

2. RELATED WORK

This chapter explores the background research and technologies that form the foundation of the Smart Glasses System. The project integrates multiple fields—computer vision, indoor navigation, wearable hardware, natural language processing, intelligent speech interaction, smart home automation, and multimodal user interfaces. To provide a comprehensive review, this chapter examines existing systems, identifies their limitations, evaluates competitor technologies, and highlights the gaps addressed by the proposed solution.

2.1. Existing Systems

Existing wearable and smart assistant technologies focus primarily on isolated functionalities. They offer speech interaction or augmented reality, but few combine multimodal AI capabilities in one consistent and user-friendly system.

2.1.1 Amazon Echo Frames (3rd Gen)

- Smart audio glasses focused on hands-free Alexa interaction.
- Provide notifications, reminders, and basic speech commands.
- No camera → therefore no computer vision or object detection.
- No indoor navigation capabilities.
- No AR overlays.
- Limited AI reasoning (relies only on Alexa cloud).
- No smart home interface on the glasses themselves.

Limitations:
Echo Frames do not support visual understanding of the environment, AR/VR interaction, indoor routing, or performing multimodal tasks.

2.1.2 Google Glass Enterprise Edition

- Provides AR overlays and hands-free interaction.

- Limited object detection and visual processing.
- Primarily built for enterprise environments (factories, warehouses).
- No built-in indoor navigation.
- Expensive hardware with limited accessibility for students or consumers.

Limitations:
Although Google Glass introduced AR, the device lacks AI reasoning, multimodal understanding, smart home integration, and indoor navigation.

2.1.3 Ray-Ban Meta Smart Glasses (2024)

- Meta AI integration
- Voice-controlled assistant
- Built-in camera
- Live streaming support
- No indoor GPS or pathfinding
- No smart home dashboard
- No VR interaction
- Weak NLP action planning

Limitations:
Meta Smart Glasses provide strong social media integration but have no multimodal AI pipeline like the proposed system.

2.1.4 Indoor Navigation Apps (Google Maps Indoor, HERE Indoor Positioning)

- Provide floor-level guidance in commercial spaces.
- Require Wi-Fi fingerprinting or BLE beacons.
- Not suitable for custom buildings without dedicated infrastructure.

Limitations:
They cannot be embedded into wearable glasses and require expensive sensor setups.

2.1.5 Smart Home Systems (Google Home, Alexa, HomeKit)

- Rely mainly on voice commands.
- No wearable integration.
- No AR or VR control interfaces.
- No customized device control or dashboards.

2.2. Overall Problems of Existing Systems

Across all competitors, several major limitations emerge:

2.2.1 Lack of Multimodal Interaction

Most systems either:

- Use speech only (Echo Frames)
- Use AR only (Google Glass)
- Or use AI only (Meta Glasses)

No existing system combines:
✔ Speech → Text → LLM → Action → AR/VR
✔ Real-time vision understanding
✔ Indoor navigation
✔ Smart home control
✔ Mobile + Web dashboards

2.2.2 No Indoor Navigation

Existing glasses rely on GPS, which:

- Fails indoors
- Cannot detect stairs/elevators/rooms
- Cannot be customized for small buildings
- This project solves the problem by:
- Creating a manual building map
- Converting it to a graph representation

- Using JSON path planning
- Supporting stairs, elevators, and corridors

2.2.3 Limited Hardware Capabilities

Market devices lack:

- Affordable microcontrollers
- Customizable sensors
- Full integration with apps
- Real-time speech processing locally

The proposed system uses:

- ESP32 with WiFi/Bluetooth
- Microphone + speakers
- On-device fast communication
- Direct connection with app and web

2.2.4 Weak NLP Reasoning

Most competitors do NOT turn user commands into structured actions.

Our project uses:

- MCB NLP pipeline
- Converts natural commands → steps → JSON → model execution
- Works with speech, vision, navigation, and smart home

2.2.5 No Computer Vision

Competitors rarely include:

- YOLO object detection
- Real-time local inference

- Custom dataset training

Our system uses:

- YOLO11
- Custom dataset
- Real-time feature extraction
- Object detection integrated with navigation

2.3. Comparison Between Existing and Proposed Method

Table 2.1: Comparison of methods

Feature	Amazon Echo Frames	Google Glass EE	Meta Glasses	Indoor Nav Apps	Proposed Smart Glasses
Computer Vision	✗	✗ Limited	✔ Basic	✗	✔ YOLO11 (custom)
Indoor Navigation	✗	✗	✗	✔ Only app-based	✔ Full indoor routing
Hardware Control	✗	✗	✗	✗	✔ ESP32 integration
Smart Home	Limited via Alexa	✗	✗	✗	✔ Full dashboard
Speech Recognition	✔ Alexa	✔ Basic	✔ Meta AI	✔	✔ Whisper multilingual
Text-to-Speech	✔	✔	✔	✗	✔ Bilingual
LLM Reasoning	Limited	✗	Moderate	✗	✔ MCB + LLM pipeline
AR Interaction	✗	✔	✗	✗	✔ AR / VR modes
Customization	Very limited	Limited	Very limited	Medium	✔ Full customization

⭐ Additional Detailed Sections You Requested

Below are the sub-sections that must appear inside Related Work because they directly support the technology foundation.

2.4 **Computer Vision in Wearable Systems**

Computer vision has been used in mobile and robotics systems, but rarely in smart glasses. Using **YOLO11**, the proposed solution provides:

- Feature extraction
- Object detection
- Identifying obstacles
- Enhancing indoor navigation

This improves:

- Safety
- Context awareness
- Accessibility for people of determination

2.5 Indoor Navigation Using Graph-Based Mapping

Traditional GPS fails indoors due to:

- No satellite visibility
- Weak signal penetration
- Multipath interference

Existing research shows that indoor navigation requires:

- Graph-based modeling
- Manual map digitization
- JSON-based routing
- Customized building layouts

Our system maps **Building 2 (Computer Science Building)** manually, converts rooms, stairs, and elevators into nodes and edges, and uses it for indoor navigation.

2.6 Hardware in Smart Glasses

Existing systems use expensive hardware.
By contrast, ESP32 offers:

- Low cost
- WiFi/Bluetooth
- Compact size
- Easy integration with glasses
- Real-time communication

Microphone + speakers allow:

- Real-time speech capture
- Playback of TTS responses
- Audio feedback for navigation

2.7 Speech Recognition (Whisper) in Wearables

Whisper is used because:

- It supports Arabic and English
- High accuracy
- Can transcribe audio/video
- Converts speech to .txt file
- Works offline or online

This enables:

- Hands-free interaction
- Fast processing
- Reliable commands for LLM

2.8 Text-to-Speech Systems

TTS research shows bilingual support improves accessibility.
Our system uses two voices:

- Arabic TTS
- English TTS

Smart Glasses decide the output voice depending on LLM language.

2.9 Large Language Models

LLMs like GPT and LLaMA transform commands into structured actions.
They provide:

- Context-aware reasoning
- Intelligent decision making
- Multi-step interpretation

This is enhanced by the **MCB NLP pipeline**, which performs:

- Intent classification
- Task decomposition
- Action-to-JSON transformation
- Model execution routing

2.10 Smart Home Integration

Existing smart homes lack wearable integration.

Our system introduces:

- Mobile app dashboard
- Web dashboard
- Device management
- Controllers
- Logs
- Statistics
- AR/VR interaction

This unifies all control methods.

2.11 User Flow and Multimodal Interaction

Compared to existing products, our system includes:

- Natural speech → text → commands
- Vision-based alerts
- AR navigation
- App dashboards
- Web dashboards

This creates a consistent and intuitive experience.

3. METHODOLOGY

This section tells how you conducted your project. It should be detailed enough to guide someone who wants to reproduce your study.

Consult your supervisor to choose only one of the sub-section groups to implement your report!

- For Software Intensive Projects;

3.1. Requirement Analysis

3.1.1 Textual Requirements

Functional Requirements

1. **Real-time Object Detection**
   - The system must detect objects using a custom-trained YOLO model.
   - The model should identify indoor landmarks (doors, elevators, room numbers, stairs, etc.).
2. **Indoor Navigation**
   - The system must compute optimal paths inside Building 2 using a custom indoor map.
   - The system should provide turn-by-turn navigation visually in AR and via speech.
3. **Multimodal Speech Interaction**
   - The glasses must capture audio and run Whisper for speech-to-text.
   - The extracted text should be processed by an LLM to provide intelligent responses.
4. **Smart Home & IoT Control**
   - Users must be able to control home devices (lights, AC, door lock) through voice.
   - The ESP32 microcontroller must receive commands over Wi-Fi and trigger actuators.
5. **Multilingual Communication**
   - The system must transcribe, translate, and generate speech output in real time.
6. **Mixed Reality Rendering**
   - The AR module must render virtual arrows, labels, and icons anchored to detected objects.
7. **User-Friendly Interaction**
   - Users can interact with the system through voice, gestures, or UI element

Non-Functional Requirements

1. **Performance**
   - Object detection must run at ≥ 20 FPS.
   - Speech recognition latency must be ≤ 2 seconds.
2. **Accuracy**
   - YOLO model accuracy must exceed 90% mAP.
   - Whisper transcription accuracy must exceed 95% for English and Arabic.
3. **Usability**
   - The interface must be simple for visually impaired and hearing-impaired users.
4. **Reliability**
   - The system should maintain stable operation for long sessions.
5. **Security**
   - Device commands must be encrypted before being sent to the ESP32 module.
6. **Portability**
   - The system should run on mobile devices and AR glasses.

3.1.2 Use Case Diagram

3.1.3 Use Case Scenarios

UC1 – Real-Time Object Detection

Step	Description
1	User starts camera mode.
2	YOLO processes the current frame.
3	Model returns detected objects + bounding boxes.
4	AR module displays labels and icons.

UC2 – Indoor Navigation

Step	Description
1	User gives voice command: “Guide me to Room 215.”
2	Whisper → converts speech to text.
3	LLM → extracts destination + context.
4	Navigation engine computes shortest route.
5	AR overlay provides arrows + callouts.

UC3 – Smart Home Control

Step	Description
1	User says: “Turn on the lights.”
2	STT module transcribes the request.
3	LLM interprets command and creates a JSON instruction.
4	System sends command to ESP32.
5	Device activates the actuator.

3.2. Design

3.2.1. Activity Diagram

3.2.2. Class Diagram

3.2.3. Deployment Diagram (and other UML diagrams if necessary)

3.3. Implementation

Main Technologies Used

- YOLOv11 for object detection
- OpenCV for camera streaming
- Whisper for speech recognition
- LLM API (OpenAI / local model)
- Unity / ARCore / ARKit for mixed reality rendering
- Python + Node.js backend
- ESP32 with MicroPython for IoT devices

3.4. Testing

- For Data/Model-driven Research Projects;

3.1. Overview of the Dataset/Model

3.2. Tools and Technology

3.3. Proposed Approach

- For System Design Projects;

3.1. Design Overview

3.2. System Architecture

The system follows a **modular architecture** consisting of six main modules:

1. **Vision Module (YOLO11 + Custom Dataset)**
2. **Indoor Navigation Module (GPS + JSON-based Floor Map)**
3. **NLP Pipeline (Whisper + LLM + Task Breakdown)**
4. **Mixed Reality Interface (AR Overlay + VR Simulation)**
5. **Smart Home Control Module**
6. **Hardware Integration Layer (ESP32 + Sensors + Battery Unit)**

Each module is loosely coupled and communicates through a central processing unit running the LLM and control logic.

3.3 Data Collection and Preparation

1. Vision Dataset Collection

- Dataset: custom images captured inside the Computer Science building.
- Classes: doors, stairs, elevators, signs, obstacles.
- Annotation Tool: Roboflow.
- Preprocessing: resizing, normalization, augmentation.

2. Floor Map Data for Indoor Navigation

- The building map was manually drawn.
- Converted into **JSON format** where each room, corridor, node, and path is represented by IDs.
- Applied graph representation for navigation (nodes, edges, weights).

3. Speech Dataset

- Whisper handles multilingual transcription.
- No manual data collection required.

3.4 Computer Vision Module

YOLO11 Model Training

We used YOLO11 due to its high speed, light weight, and strong accuracy in real-time environments.

Steps:

1. Annotate images.
2. Split into train/val/test.
3. Train using transfer learning.
4. Validate accuracy (mAP, precision, recall).
5. Export ONNX version for real-time inference.

Why We Chose YOLO11

- Outperforms YOLOv8 in speed.
- Lightweight enough to run on portable devices.
- Excellent for object detection in dynamic environments.

3.5 Indoor Navigation Module

Creating the Navigation Graph

- Each corridor, hallway, and room is converted into nodes.
- Connections between nodes form edges with distances as weights.

- The result: a navigable graph.

Pathfinding Algorithm

- We apply A* because:
- It is faster than Dijkstra.
- Provides optimal paths.
- Works well for indoor map constraints.

3.6 NLP + LLM Reasoning Module

The NLP pipeline handles user speech, understands intent, breaks tasks into steps, and triggers actions.

Pipeline Flow

1. Whisper converts speech → text.
2. LLM analyzes the text.
3. Task Breakdown Engine (MCB) converts the sentence into sub-steps.
4. Action Manager decides what to trigger (navigation, smart home, CV, AR overlay).
5. Output JSON defines system actions.

Why LLM?

- Understands natural language.
- Can perform reasoning.
- Handles complex commands like:

"Guide me to the AI lab and tell me if there are stairs on the way."

3.7 Mixed Reality (AR/VR Module)

AR Overlay

- Arrows are shown on the lenses.
- Based on navigation output.

- Highlights detected objects with bounding boxes.

VR Mode

Used for testing and for users with disabilities to simulate navigation.

3.8 Smart Home Automation Module

The smart glasses communicate with a Flask/FASTAPI backend.

Supported Features:

- Turn lights on/off
- Open/close smart door
- Adjust temperature
- Control appliances

3.9 Hardware Design and Integration

Components Used:

- ESP32 microcontroller
- Bone conduction speakers
- Dual microphones
- Ultra-light camera
- Rechargeable battery pack
- Bluetooth/WiFi module

Why This Hardware

- Low power consumption
- Lightweight
- Supports real-time streaming
- Affordable for a graduation project

3.10 System Workflow Summary

1. User speaks command.
2. Whisper transcribes it.
3. LLM interprets and classifies the task.
4. CV module or Navigation module activates.
5. Output is displayed through AR overlay.

Smart home actions executed if needed.

3.11 Implementation Steps (Detailed)

1. **Collect Datasets (Vision + Map)**
    - Identify the types of data required for the project: images for computer vision and maps for indoor navigation.
    - For vision datasets: collect images of objects, environments, or specific locations relevant to the use case. Include diverse conditions such as lighting variations, occlusions, and multiple angles to ensure model generalization.
    - For navigation datasets: obtain floor plans of buildings, including corridors, stairs, elevators, entrances, and exit points. If indoor GPS is not available, use manual measurements or mapping tools.
    - Ensure the dataset is comprehensive and representative to avoid bias in model performance.
2. **Prepare and Annotate Images**
    - Clean and preprocess all collected images by resizing, normalizing, and correcting orientations.
    - Annotate objects using professional labeling tools (e.g., LabelImg, CVAT). Create bounding boxes, masks, or keypoints depending on the YOLO model requirements.
    - Maintain annotation consistency and define a clear labeling convention to reduce errors.
    - Split the dataset into **training (70%)**, **validation (15%)**, and **test (15%)** sets to ensure accurate model evaluation.
3. **Train YOLOv11 Model**
    - Configure YOLOv11 architecture: define number of classes, anchor boxes, input dimensions, and backbone network.
    - Prepare the data pipeline to feed annotated images into the model efficiently, including data augmentation (rotation, scaling, flipping) to improve robustness.

- Train the model on a GPU-enabled environment to speed up learning. Monitor training loss, precision, recall, and mAP (mean Average Precision) metrics.
- Perform hyperparameter tuning such as learning rate, batch size, and epochs for optimal performance.
- Test the trained model on the test set to evaluate generalization and fine-tune if necessary.

4. **Build Navigation Graph + JSON Map**
   - Convert building floor plans into a **graph structure** where nodes represent key locations (rooms, intersections) and edges represent paths or connections.
   - Include details such as stairs, elevators, and corridors. Assign weights to edges to indicate distance or difficulty.
   - Encode the graph in **JSON format**, storing all nodes, edges, and metadata to allow the system to parse and compute navigation routes programmatically.
   - Implement pathfinding algorithms (e.g., Dijkstra or A*) to calculate the shortest and most efficient path to the target destination.
5. **Develop Whisper + LLM Pipeline**
   - Integrate the **Whisper speech-to-text engine** to process audio inputs from the user.
   - Detect the spoken language (Arabic or English) automatically and transcribe the speech accurately.
   - Send the transcribed text to the **Large Language Model (LLM)** for semantic understanding and task planning.
   - Translate user commands into structured steps in **JSON format** for downstream execution by different modules (navigation, AR, smart home control).
   - Ensure error handling for misheard commands, ambiguous instructions, or unsupported requests.
6. **Implement AR Rendering**
   - Develop the AR engine to overlay digital objects, navigation indicators, and notifications onto the real-world view captured by the glasses.
   - Integrate outputs from YOLO, GPS/IPS navigation, and environmental sensors to render accurate, real-time information.
   - Optimize frame rate and latency for smooth visualization in the Heads-Up Display (HUD).
   - Test AR overlays in multiple lighting and environmental conditions to ensure stability and reliability.
7. **Integrate ESP32 Hardware**
   - Connect microphone, speaker, sensors, and other peripherals to the **ESP32 microcontroller**.
   - Establish wireless communication via **Wi-Fi or Bluetooth** between ESP32 and Smart Glasses for real-time data transfer.
   - Implement firmware to handle audio capture, playback, sensor reading, and command execution.

- Ensure the hardware responds instantly to commands from both the user and the software modules.

8. **Test Modules Individually**
    - **Vision Module:** Evaluate object detection accuracy and robustness using test images.
    - **Navigation Module:** Verify route calculation and indoor/outdoor navigation correctness.
    - **Whisper + LLM Pipeline:** Check transcription and command interpretation accuracy.
    - **AR Rendering Module:** Test overlay placement, HUD clarity, and AR responsiveness.
    - **Hardware Module:** Confirm microphone input, speaker output, and sensor readings are accurate.
    - Record results and debug any issues before integrating modules.
9. **Combine Modules for Full System**
    - Integrate all modules into a single workflow to ensure seamless interaction between hardware and software.
    - Test communication between modules: commands from LLM should trigger navigation, AR, smart home actions, or media capture correctly.
    - Implement synchronization mechanisms and handle concurrency issues to prevent system crashes.
    - Perform end-to-end testing to ensure system stability and reliability under different use scenarios.
10. **Evaluate Performance**
    - Define performance metrics:
        - YOLO detection accuracy
        - Speech recognition accuracy
        - Navigation precision
        - AR overlay responsiveness
        - Smart home command execution latency
    - Conduct real-world testing in multiple environments.
    - Identify bottlenecks and optimize system components where necessary.
    - Document results with screenshots, logs, and statistics to support evaluation.

3.3. System Software

4. EXPERIMENTAL RESULTS

The following points should be considered when presenting your findings:

- Present your findings in a clear and easy-to-understand manner.
- Consider your readers; make it easy for them to understand the data.
- Include only the particularly important findings in the body of the graduate report. Do not distract the reader with very detailed data. If you have very detailed information that you would like the reader to refer to, consider including it in an appendix. Remember to refer the reader to the appendix.
- Consider the most effective presentation style for your results. Normally a combination of text and tables/figures is the preferred style. Tables and figures provide data in numeric or pictorial terms in a more visual manner than straight text. The straight text, however, enables you to explain the significance of the data. The straight text also enhances the fluency of the chapter and helps the reader to focus on the most important aspects of the data.
- Ensure that your tables and figures add more information than that given in the text. Do not just display visually what has already been described.

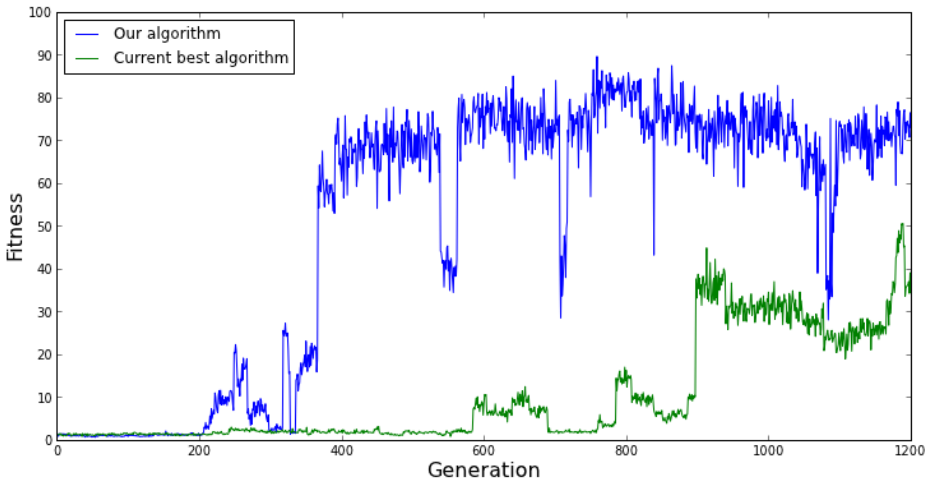

Figure 4.1: Comparison with the current best algorithm and our algorithm

5. DISCUSSION

In this section, you should restate the problem to address, and summarize how the results have addressed it. Students should discuss the significance of all the results, and interpret their meaning. Potential sources of error should be discussed, and anomalies analyzed. Finally, you should tie your conclusions into the "big picture" by suggesting the impact and applications this research might have. This can be accomplished by discussing how the results of this project will affect the project's domain, what future experiments could be carried out based on this research, or what affect the conclusions could have on the industry.

6. CONCLUSIONS

State a brief summary of your interpretations and conclusions regarding the project's topic. Recommended structure moves from Specific to General;

- Begins with a reiteration of the project topic (tone is emphatic),
- May summarize main points and findings,
- Brings the topic back to some general significance or relevance,
- Finally, provides future directions to this study.

REFERENCES

Every citation made in the body of the project report must appear in the References. Similarly, every item listed in the References must be cited in the body of the report. Follow the APA standard method for citing and listing both the print references and online references. Examples;

[1] Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. Fuzzy sets and systems, 1(1), 3-28.

[2] Mitchell, J.A., Thomson, M., & Coyne, R.P. (2017). A guide to citation. London, England: My Publisher

[3] Troy, B.N. (2015). APA citation rules. In S.T, Williams (Ed.). A guide to citation rules (2nd ed., pp. 50-95). New York, NY: Publishers.

[4] Rosten, E., & Drummond, T. (2006, May). Machine learning for high-speed corner detection. In European conference on computer vision (pp. 430-443). Springer, Berlin, Heidelberg.

[5] Fowler, M., & Lewis, J. (2014). Microservices a definition of this new architectural term. URL: http://martinfowler.com/articles/microservices.html

!Hint: You may use Google Scholar to create APA style references in an easy way.

APPENDIX

Include additional content (raw data, code listing, etc.) as necessary to provide a detailed explanation that is not essential in the body of the report but that would be of interest of readers. If this section is not used, remove it from the project template. In case of having multiple appendix sections, informatively title and label as Appendix A, Appendix B, etc., according to the order in which they are mentioned in the text.