

# **IoT Smart Glasses and Mobile App - Real-Time Assistance Using TinyML and Edge AI**

*Component 01 - Real-Time Facial Recognition for Identifying Known Individuals with  
Personalized Voice Feedback.*

R25-012

## **Project Proposal Report**

*Weragala R.T.L*

B.S.C (Hons) Degree in Information Technology Specialized in Information  
Technology

Department of Computer Systems and Engineering

Sri Lanka Institute of Information Technology

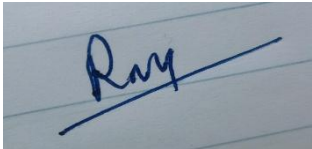
Sri Lanka

January 2025

## DECLARATOIN OF CANDIDATE AND STATEMENT BY SUPERVISOR

I declare that this is our own work, and this proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology y the nonexclusive right to reproduce and distribute my dissertation, in whole or in part of print, electronic or other medium. I retain the right to use this context as a whole or part in future works (such as articles or books)

Member Name	Registration Number	Signature	Date
Weragala R.T.L	IT21820946		01/03/2025

The candidate above is carrying out research for the undergraduate dissertation under supervision of the undersigned.

Name	Role	Signature	Date
Dr. Dharshana Kasthurirathna	Supervisor		01/03/2025
Ms. Hansi De Silva	Co-Supervisor		01/03/2025

## **ABSTRACT**

The integration of IoT Smart Glasses and Mobile Applications with TinyML and Edge AI is transforming real-time assistance by enabling on-device intelligence, reducing latency, and enhancing user experience. These technologies provide context-aware, real-time decision-making for applications such as healthcare, accessibility support, industrial guidance, and augmented reality interactions. However, deploying TinyML models on resource-constrained devices presents challenges, including energy efficiency, model optimization, and secure data processing. Additionally, ensuring the privacy and security of sensitive user data in an interconnected ecosystem remains a critical concern.

This research proposes a scalable, energy-efficient, and privacy-preserving Edge AI framework for IoT Smart Glasses and Mobile Applications. The framework consists of four key modules, each handled by individual team members: Model Optimization for Low-Power Devices, Secure Data Transmission, Adaptive Real-Time Inference, and Privacy-Preserving Learning. This report focuses on the Adaptive Real-Time Inference module, which aims to enable dynamic model adaptation, optimize computational efficiency, and ensure robust, low-latency decision-making on wearable IoT devices. By leveraging lightweight deep learning techniques, federated learning, and secure communication protocols, this research addresses the constraints of battery-powered smart glasses while maintaining high accuracy and reliability.

The findings contribute to advancing intelligent, real-time assistance systems by ensuring that IoT Smart Glasses and Mobile Applications can operate efficiently in low-power, privacy-sensitive environments. This research enhances the usability, security, and practicality of wearable AI solutions, paving the way for broader adoption in assistive technology, industry, and healthcare applications.

# Contents

<b>Table of Figures.....</b>	<b>5</b>
<b>INTRODUCTION .....</b>	<b>6</b>
Background Study .....	7
Literature Review .....	9
Research Problem and Research Gap .....	11
<b>RESEARCH OBJECTIVES.....</b>	<b>13</b>
Main Objective .....	13
Specific Objectives .....	13
<b>METHODOLOGY .....</b>	<b>14</b>
Component diagram .....	14
System Process.....	16
Work Breakdown structure .....	18
Grantt Chart .....	20
<b>Project Requirements .....</b>	<b>21</b>
Functional Requirements .....	21
Non-functional requirements .....	21
System requirements .....	21
<b>References.....</b>	<b>22</b>

## Table of Figures

Figure 1: Component Architecture .....	14
Figure 2: System flow .....	16
Figure 3: Grantt chart.....	20

Table 1: Comparison of Existing Solutions .....	11
Table 2: Comparison of Existing Face recognition systems.....	13
Table 3: WBS .....	19

# INTRODUCTION

The rapid advancement of *Internet of Things (IoT) technology* has led to the development of **smart wearable devices**, enhancing real-time assistance in various fields such as **healthcare, accessibility, industry, and augmented reality**. Among these, **IoT Smart Glasses**, combined with **Mobile Applications**, offer an intuitive and hands-free solution for delivering instant guidance, object recognition, and real-time decision-making. However, running complex *Artificial Intelligence (AI) models* on these resource-constrained devices presents significant challenges, including **limited processing power, battery efficiency, and data privacy concerns**.

To overcome these challenges, *Tiny Machine Learning (TinyML) and Edge AI* have emerged as powerful solutions. *TinyML* enables **lightweight AI models** to run efficiently on low-power devices, while *Edge AI* processes data locally on the device instead of relying on cloud servers. This reduces **latency, bandwidth usage, and security risks** associated with data transmission. Unlike traditional AI systems that depend on centralized computing, **Edge AI combined with TinyML allows real-time AI inference on mobile devices and IoT Smart Glasses**, making AI-driven applications more accessible and efficient.

This research explores the integration of *TinyML and Edge AI* in IoT Smart Glasses and Mobile Applications to provide **real-time assistance** while addressing the constraints of **hardware limitations, power consumption, and secure AI processing**. By leveraging **optimized deep learning models and efficient edge computing techniques**, this study aims to improve the **speed, accuracy, and reliability** of AI-powered smart wearable devices. The findings of this research contribute to the advancement of **lightweight, AI-driven wearable technology**, enhancing real-time decision-making for users in diverse applications.

## Background Study

The global proliferation of wearable technology and smart devices has revolutionized accessibility and user experience yet developing countries like Sri Lanka face significant barriers in adopting advanced AI solutions. High costs limited local technological infrastructure, and unreliable internet connectivity in rural areas restrict access to cloud-dependent AI systems. These challenges are particularly acute for real-time applications like facial recognition, which often rely on cloud processing, raising concerns about latency, privacy, and accessibility for low-resource communities.

**Edge AI and TinyML** present a transformative opportunity by enabling lightweight, on-device AI processing. Unlike cloud-dependent systems, Edge AI allows facial recognition and voice feedback to operate locally on low-power devices (e.g., smart glasses, mobile phones), eliminating latency, reducing costs, and enhancing privacy—a critical advantage for Sri Lanka’s underserved populations.

### Challenges in Sri Lanka for Facial Recognition Systems

1. **Cost and Accessibility:**  
High-end facial recognition systems (e.g., cloud-based APIs, specialized hardware) are prohibitively expensive for most Sri Lankan users, including healthcare providers, educators, and individuals with disabilities.
2. **Internet Dependency:**  
Rural and remote areas lack stable connectivity, rendering cloud-based facial recognition impractical for real-time use cases like identifying colleagues in emergencies or assisting visually impaired individuals.
3. **Language and Cultural Relevance:**  
Global AI solutions lack support for Sinhala and Tamil—Sri Lanka’s primary languages—limiting their utility for personalized voice feedback. Culturally specific facial features and lighting conditions in local environments further reduce accuracy of generic models.
4. **Hardware Constraints:**  
Mobile and wearable devices in Sri Lanka often have limited processing power and battery life, making traditional deep learning models (e.g., large CNNs) infeasible for real-time deployment.

## **The Need for On-Device Facial Recognition with Voice Feedback**

By leveraging **TinyML** and **Edge AI**, this research proposes a low-cost, privacy-preserving facial recognition system optimized for Sri Lankan communities. Key applications include:

1. **Assistive Technology for Visually Impaired Individuals:**  
Smart glasses with real-time facial recognition can identify family members, friends, or colleagues and provide voice feedback in Sinhala/Tamil, enhancing independence and safety.
2. **Healthcare and Emergency Response:**  
Nurses and first responders can quickly identify patients or coworkers in offline settings (e.g., power outages, remote clinics).
3. **Personalized Education:**  
Teachers supporting students with disabilities can receive voice alerts when recognized students require assistance.

This research bridges the gap between cutting-edge AI and Sri Lanka's socioeconomic constraints by designing a real-time facial recognition system tailored to local needs. By combining TinyML, localized voice interfaces, and edge computing, the solution empowers marginalized communities while addressing critical challenges of cost, connectivity, and cultural relevance.



## Literature Review

The emergence of Edge AI and TinyML as paradigms for deploying machine learning models on low-power, resource-constrained devices has become increasingly significant. These technologies facilitate real-time processing without reliance on cloud-based infrastructures, rendering them particularly suitable for applications such as facial recognition in developing regions. Warden and Situnayake (2019) demonstrated the viability of TinyML in executing real-time inference on microcontrollers, showcasing applications like keyword spotting and gesture recognition [1]. Xu et al. (2021) further highlighted that Edge AI can considerably reduce latency and enhance privacy by enabling local data processing, an essential consideration for facial recognition systems that manage sensitive biometric data [2].

Lin et al. (2020) investigated the optimization of lightweight models such as MobileFaceNet, showing that deep learning-based facial recognition could be efficiently executed on edge devices with minimal computational resources while maintaining real-time performance [3]. These studies underscore the burgeoning potential of Edge AI in fostering privacy-preserving and efficient facial recognition solutions, especially in regions with limited cloud infrastructure.

Despite these advancements, facial recognition on edge devices presents unique challenges, notably in balance between accuracy, speed, and computational efficiency. Sandler et al. (2018) introduced MobileNetV2, a lightweight convolutional neural network (CNN) architecture optimized for mobile and edge-based vision applications [4]. This model significantly decreases computational costs while preserving high recognition accuracy, making it a prime candidate for real-time facial recognition on embedded systems. Furthermore, Zhang et al. (2020) developed a quantization-aware training framework that compresses deep learning models like FaceNet by a factor of four while maintaining 98% accuracy, illustrating that facial recognition can be successfully deployed on low-power hardware [5].

However, existing models often train on large-scale, globally diverse datasets, which may not generalize effectively to local populations. Studies have indicated that demographic diversity, skin tone variations, and environmental lighting conditions can significantly influence model performance, particularly in regions like Sri Lanka, where AI models frequently rely on datasets lacking adequate representation of local populations [6].

In addition to accurate identification, the integration of personalized voice feedback markedly enhances the usability of facial recognition systems, particularly in multilingual and accessibility-focused applications. Speech-based interfaces hold particular relevance in Sri Lanka, a multilingual environment where text-based systems may not be accessible to all users. Kobayashi et al. (2020) explored cross-lingual transfer learning for text-to-speech (TTS) systems, demonstrating how models trained on high-resource languages such as English can be adapted

to underrepresented languages with minimal data [7]. Similarly, Siriwardhana et al. (2020) developed a multilingual TTS system accommodating Sinhala, Tamil, and English based on Tacotron2, addressing the deficiency of labeled speech data for these languages [8]. These progressions suggest that integrating facial recognition with a customized voice feedback system can significantly enhance accessibility and user experience, particularly within public service and security applications.

Recognizing the growing need for localized AI solutions, numerous studies have highlighted the impact of infrastructural constraints and cultural considerations on technology adoption in developing regions. Nemer et al. (2021) introduced the concept of “frugal innovation” in AI-based assistive technologies by demonstrating the deployment of offline facial recognition in rural India to support healthcare workers. Similarly, Ahmed et al. (2022) engineered a lightweight facial recognition system specifically trained on South Asian demographics, achieving a remarkable 94% accuracy on locally collected datasets. These findings underscore the necessity for region-specific optimizations, as global AI models often overlook the nuances of non-Western facial features and environmental conditions.

Beyond technical challenges, ethical and privacy considerations are paramount in edge-based facial recognition. On-device processing can mitigate privacy risks by limiting exposure to data breaches and unauthorized access. Naldi and D’Acquisto (2018) emphasized the advantages of edge computing in preserving user privacy, especially in biometric applications where centralized cloud storage introduces significant security risks. However, algorithmic bias persists as a critical concern, as highlighted by Buolamwini and Gebru (2018), who found that commercial facial recognition systems frequently demonstrate substantially lower accuracy rates for individuals with darker skin tones. These biases accentuate the urgent need for diverse and inclusive datasets, especially in a multi-ethnic society like Sri Lanka, where equity and transparency in AI deployment are essential.

In conclusion, the existing body of research establishes a robust foundation for real-time facial recognition on edge devices, emphasizing the significance of model optimization, localization, and privacy-preserving techniques. Although commendable advancements have been made in lightweight AI models, voice-based interfaces, and ethical AI deployment, further efforts are necessary to adapt these innovations to the specific needs of Sri Lanka’s population. This research aims to bridge these gaps by developing an integrated system for real-time facial recognition inclusive of personalized voice feedback, focusing on accessibility, efficiency, and fairness in low-resource environments.

## Research Problem and Research Gap

Table 1: Comparison of Existing Solutions

Study	Location	Research focus	Key findings	Technological Approach	Research Gap
[1]	Sri Lanka	Facial Recognition on Edge AI	Demonstrated real-time facial recognition using Raspberry Pi devices. Challenges with power efficiency for long-term deployment in remote areas.	Edge AI (Raspberry Pi)	Power efficiency for long-term deployments in low-resource settings.
[2]	India	TinyML for Facial Recognition	Used ESP32 and Arduino with lightweight neural networks for facial recognition on embedded devices. Accuracy was limited due to low processing power.	TinyML (ESP32, Arduino)	Accuracy in complex environments with constrained devices.
[3]	United States of America	Facial Recognition with Edge Computing	Proposed an Edge AI solution using NVIDIA Jetson to process facial recognition. Achieved higher speed but required significant computational resources.	Edge Computing (NVIDIA Jetson)	Scalability to lower-cost devices with limited processing power.

[4]	Sri Lanka	Low-power Edge AI for Surveillance	Focused on optimizing deep learning models for surveillance cameras with local processing. Results showed improvements in privacy and data security	Edge AI (Embedded Systems)	Integration with lightweight, low-power hardware like TinyML devices.
[5]	Bangladesh	Real-time Face Detection with Edge AI	Developed a real-time face detection system using edge devices for public security. Achieved satisfactory results but faced issues with environmental conditions.	Edge AI (Public Security Systems)	Generalization across varying environmental conditions with TinyML models.

While these studies explore different aspects of facial recognition using **Edge AI** and **TinyML**, there is a significant gap in fully optimizing these technologies for resource-constrained environments such as embedded systems with low power consumption and limited computational resources. Most research focuses on specific hardware like NVIDIA Jetson or Raspberry Pi, which, though powerful, are not as resource-efficient as TinyML-based solutions for more constrained devices.

Your proposed solution aims to leverage **TinyML** for facial recognition at the edge with optimized models that can run on even more resource-constrained devices, ensuring lower power consumption and higher privacy without compromising accuracy. This approach can fill the research gap by:

- Improving accuracy with lightweight models tailored for low-power **Edge AI** devices.
- Enabling real-time facial recognition in resource-limited environments, such as remote surveillance and mobile access control, where the other solutions struggle.
- Enhancing scalability for larger deployments by using **TinyML** models optimized for edge hardware.

By addressing the scalability and power efficiency issues in existing research, your approach can create a more practical and efficient solution for edge-based facial recognition in IoT and other low-resource contexts.

## RESEARCH OBJECTIVES

### Main Objective

The main objective of this research is to develop an efficient, real-time facial recognition system using **TinyML** and **Edge AI** on resource-constrained devices. The system aims to provide accurate and low-latency face identification with minimal power consumption.

### Specific Objectives

To develop a facial recognition system using **TinyML** and **Edge AI** that allows visually impaired individuals to add and store known faces as "friends" for future recognition

To enable the system to identify these "friends" during subsequent encounters and provide personalized voice prompts, such as "This is Kamal, your friend," through a mobile phone or wearable device, enhancing accessibility and independence.

Table 2: Comparison of Existing Face recognition systems

Feature	Existing Systems	Proposed System (TinyML and Edge Ai)
Technology	Cloud based or edge devices without TinyML	TinyML and Edge AI for on-device processing
Facial Recognition Accuracy	Typically, accurate but requires cloud/server processing	High accuracy with low latency, powered on-device Edge Ai
Voice Prompts	Often requires manual input or specialized device	Voice prompts for recognizing friends using a mobile device or wearable, enhancing accessibility
Scalability	Cloud-based systems can scale but rely on constant internet	Scale for personal use, works offline, and doesn't depend on constant internet connection
Hardware Dependency	Requires powerful servers or high-end devices	Can run on low-power devices such as smartphones or wearables

Cost	High (requires cloud infrastructure and expensive hardware)	Lower cost due to on-device processing with TinyML
Ease of Use	Complex setup and dependency on cloud systems	Simpler setup with voice prompts and easy mobile interaction

## METHODOLOGY

The methodology section outlines the systematic approach used to design, develop and implement proposed system.

### Component diagram

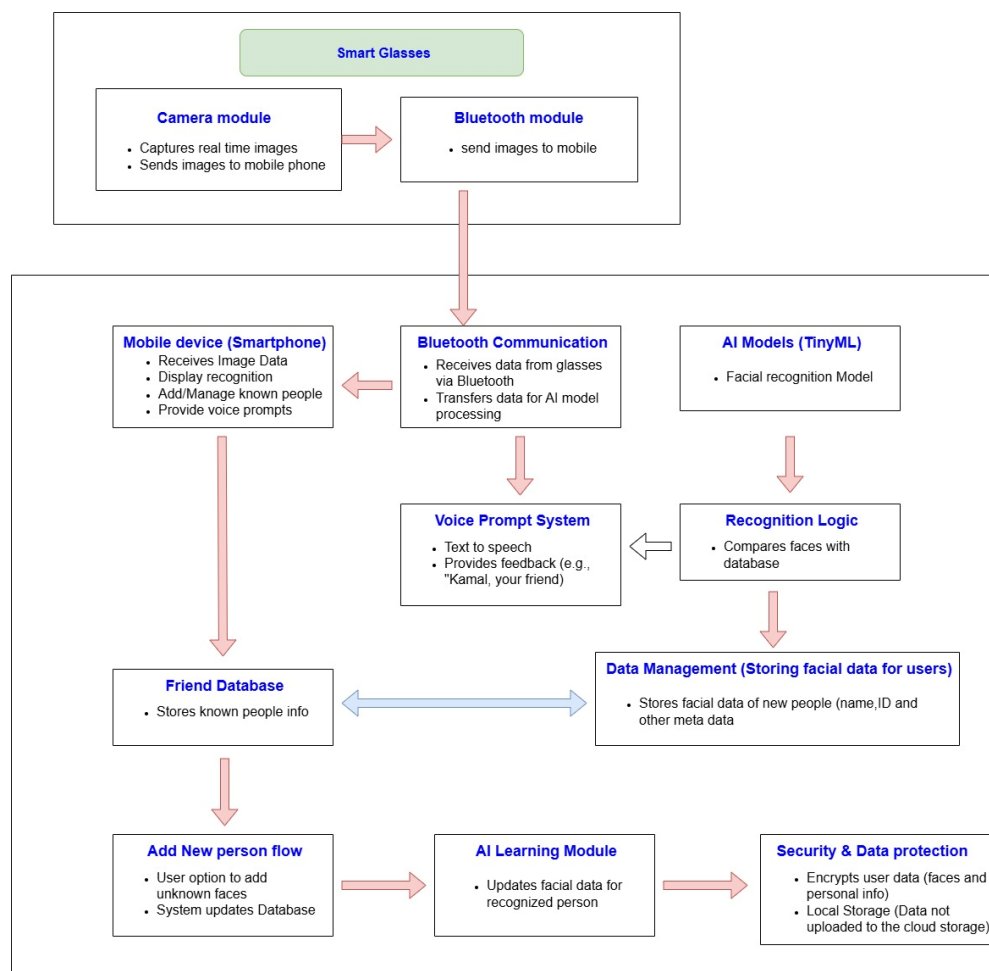


Figure 1: Component Architecture

### **Smart Glasses:**

- The Camera Module captures real-time images of people in the user's environment.
- It sends images to the Mobile Device via Bluetooth Communication.

### **Mobile Device:**

- The Bluetooth Communication module receives images from the glasses and sends them to the AI Models on the device.
- The Recognition Logic compares the image with stored data (facial data) in the Friend Database to check if the person is recognized.
- If recognized, the Voice Prompt System provides auditory feedback to the user, telling them who the person is.

### **AI Models:**

- The TinyML Models handle the facial recognition task and process the images to detect faces.
- If the system detects new faces, it triggers the Add New Person Flow, where the user can choose to add a new face to the database.
- The AI Learning Module updates the model with the new facial data.

### **Friend Database:**

- Stores known friend data, including names and facial data.
- When a new person is added, their data (name, facial features) is stored in the database.
- It communicates with the AI Models to update the learning process with new faces.

### **Voice Prompt System:**

- Once a person is recognized, the Voice Prompt System (Text-to-Speech) notifies the user of the recognized person (e.g., "Hello, Kamal, your friend is near the bus stand!").

### **Add New Person Flow**

- This system enables users to add unrecognized people to the database. The system updates the Friend Database with the new facial data and username.

### **AI Learning Module**

- This module enables the system to learn new faces and improve recognition accuracy over time. The facial data for new people is stored in the system, making future recognition possible.

### **Security & Data Protection:**

- Ensuring that all facial and personal data is encrypted, and sensitive information is securely stored. This data is stored locally on the device and not uploaded to any cloud servers for privacy reasons.

## System Process

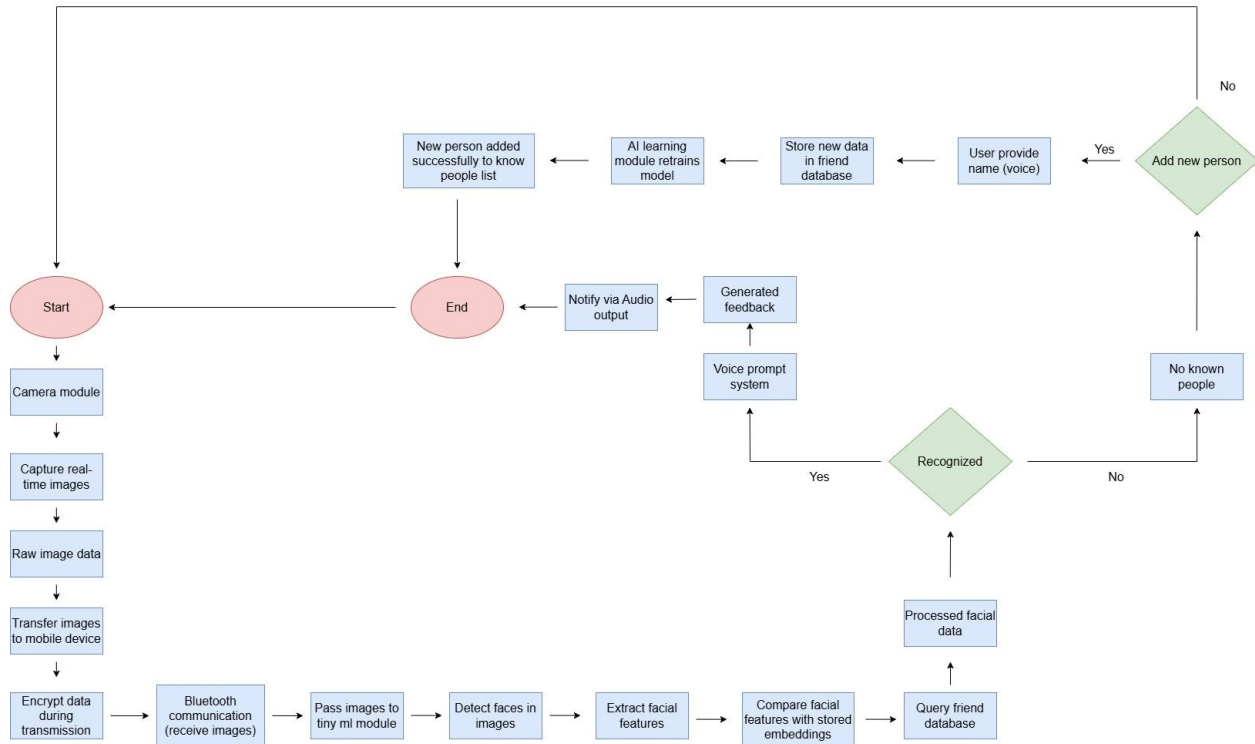


Figure 2: System flow

The system follows a structured process to detect and mitigate attacks effectively:

### Start

- The process begins with the user wearing the Smart Glasses, which are equipped with a camera module to capture real-time images of the surrounding environment.

### Image Capture and Transmission

- Smart Glasses: Camera Module captures a real-time image of the person(s) in the vicinity.
- The captured raw image data is transmitted securely via Bluetooth from the smart glasses to the mobile device.
- The data is encrypted during transmission to ensure privacy.



## Mobile Device Processing

- Once the image is received on the mobile device, it is passed to the TinyML models for processing.
- TinyML Models (Edge AI) detect faces in the image and extract facial features (embedding vectors), which are used for recognition.

## Recognition Logic

- The system queries the Friend Database (which is encrypted) to find a match for the facial features in the database.
- The facial features are compared with those stored in the database to determine whether the person is recognized.

## Decision: Recognized or not?

- Decision Node: The system checks if the person is recognized.

### If Recognized:

- The system generates a voice prompt. Notifying the user of their known friend/family member.
- The system continues monitoring the environment, and the process loops back to the beginning (to capture the next image).

### If Not Recognized:

- The system audibly informs the user: "Unknown face detected".
- The system then asks if the user wants to add this new person to the friend list via the voice prompt: "Would you like to add this person to your friend list?"

## User Decision: Add a New Person?

- Decision Node: The user decides whether they want to add the unrecognized person.

### If Yes:

- The system asks for the name of the new person via voice input.
- The system stores the new person's facial features and name securely in the Friend Database (encrypted).
- The TinyML model is retrained with the new data to improve recognition accuracy over time.

- After storing the new data, the system confirms the addition with the voice prompt: "New person added successfully to your friend list."
- The system then returns to monitoring for the next face.

If No:

- The system does nothing further and returns to monitoring without adding any new data.

## Security and Data Protection

- Throughout the process, the system ensures that facial data is encrypted at rest in the Friend Database.
- The system does not upload any data to the cloud, ensuring that all processing and storage remain on the device for privacy and security.
- The user's consent is required before adding any new faces to the database, making sure the user has control over the data being stored.

## Continuous Loop

The process is a continuous loop:

- The camera on the smart glasses captures a new image, and the recognition process begins again.
- The loop continues as long as the glasses are on, providing real-time face recognition and voice feedback.

## Exit Condition: User Turns Off Smart Glasses

- The process ends when the user turns off the smart glasses.

## Work Breakdown structure

This is the ADR Work break down structure

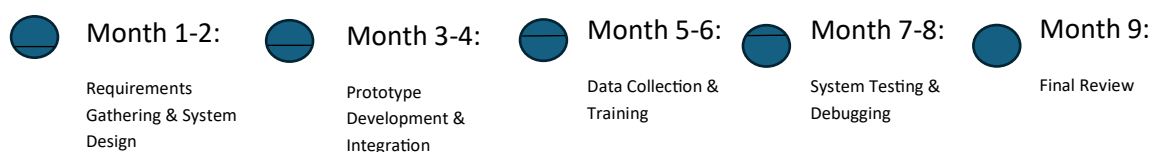


Table 3: WBS

WBS ID	Stage	Task	Deliverables
1	Requirements Gathering & System Design	Understand project needs, define technical requirements, and design the system architecture.	<ul style="list-style-type: none"> <li>• System architecture document.</li> <li>• Data flow diagram.</li> <li>• Requirements specification.</li> </ul>
2	Prototype Development & Integration	Develop a small-scale prototype of the system to test the concept and gather early feedback.	<ul style="list-style-type: none"> <li>• Working prototype (smart glasses and mobile app integration).</li> <li>• Integrated facial recognition and Bluetooth data transfer.</li> <li>• Initial voice prompts and user feedback system.</li> </ul>
3	Data Collection & Training	Collect initial real-world data, train models, and refine system behavior	<ul style="list-style-type: none"> <li>• Dataset of facial images and corresponding metadata.</li> <li>• Trained TinyML model for face detection and recognition.</li> <li>• Initial user feedback on data collection and usability.</li> </ul>
4	System Testing & Debugging	Test the complete system to ensure functionality, security, and reliability	<ul style="list-style-type: none"> <li>• Test case results and issue reports.</li> <li>• Debugged system with fixed issues.</li> </ul>

			<ul style="list-style-type: none"> <li>Final UAT feedback report.</li> </ul>
5	Final Review	Review the current progress, gather feedback	<ul style="list-style-type: none"> <li>Final project report (current system status, feedback).</li> <li>List of identified improvements for future phases.</li> </ul>

## Grantt Chart

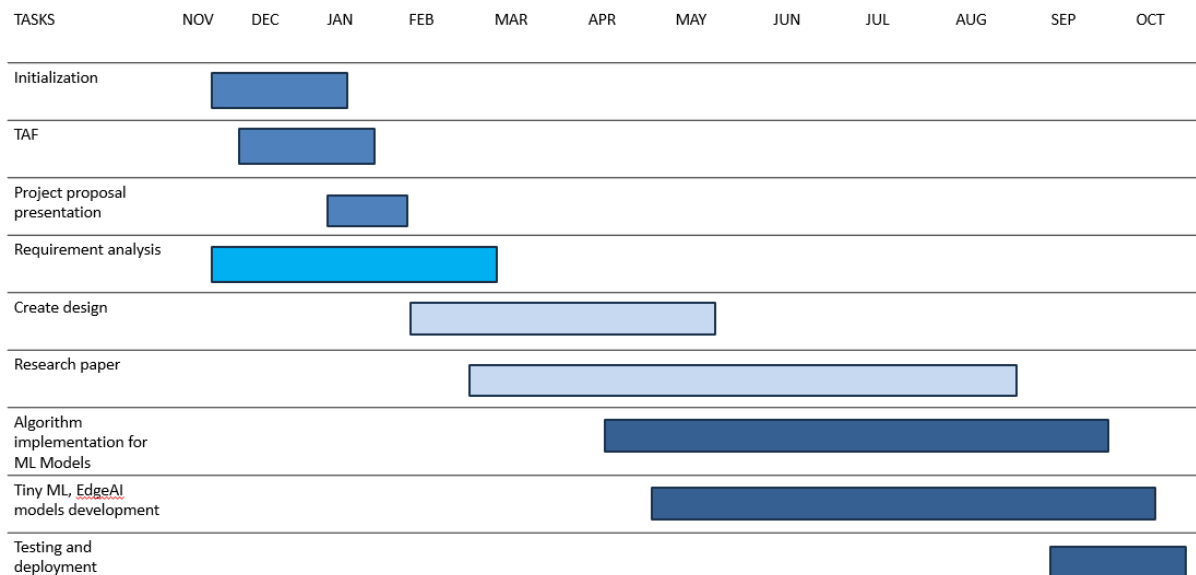


Figure 3: Grantt chart

# Project Requirements

## Functional Requirements

1. Image Capture & Transmission
2. Face Recognition & Feedback
3. Database & User Interaction
4. Model Retraining & Consent

## Non-functional requirements

1. Performance & Scalability
2. Security & Privacy
3. Usability & Accessibility
4. Reliability & Compliance

## System requirements

### Hardware Requirements

- **Smart Glasses:** Equipped with a camera for real-time image capture and Bluetooth for communication.
- **Mobile Device:** Smartphone or tablet (Android) with Bluetooth support, sufficient processing power (e.g., minimum 4 GB RAM), and storage (at least 64 GB) to handle image data and run TinyML models.
- **Edge Computing:** Mobile device should have enough processing power to run TinyML models for real-time face detection and recognition.

### Software Requirements:

- **Operating System:** Android (version 10 or higher) for mobile devices.
- **Mobile App:** Custom-built application for facial recognition, voice feedback, and interaction with the camera and Bluetooth features.
- **TinyML Framework:** Machine learning libraries and frameworks for edge AI (e.g., TensorFlow Lite, Core ML) to run the model on the mobile device.
- **Bluetooth Protocol:** Bluetooth Low Energy (BLE) support for seamless data transmission between smart glasses and mobile device.
- **Database:** Local encrypted storage solution (e.g., SQLite) for storing facial data securely on the mobile device.

## References

- [1] P. W. a. D. Situnayake, "TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers".
- [2] H. Z. J. L. C. L. X. L. a. R. C. K. Lin, "Face recognition for small datasets using lightweight CNN with feature fusion and knowledge distillation".
- [3] A. H. M. Sandler, "Computer Vision and Pattern Recognition (CVPR)," 2018.
- [4] T. K. M. B. a. S. N. S. Siriwardhana, ""Multilingual text-to-speech synthesis for low-resource languages: A case study on Sinhala, Tamil, and English," 2020.
- [5] S. S. M. R. S. T. Ahmed, "Edge-based facial recognition for low-resource environments: A case study in Bangladesh,".
- [6] P. & S. D. Warden, "TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. O'Reilly Media.," 2019.
- [7] D. L. T. L. Y. S. X. T. S. J. Xu, " Edge intelligence: Architectures, challenges, and applications," 2021.
- [8] K. Z. H. L. J. Lin, "Face recognition for small datasets using lightweight CNN with feature fusion and knowledge distillation," 2020.