1. Introduction

Wearable systems capable of recognizing human gait and posture play an important role in health monitoring, rehabilitation, sports analysis, and activity tracking. Traditionally, such systems rely on cloud-based processing, which increases latency, power consumption, and dependency on network connectivity.

With the advancement of TinyML, it has become possible to perform machine learning inference directly on low-power microcontrollers. This project focuses on designing a wearable gait and posture recognition system that performs real-time classification using IMU data and TinyML techniques, targeting deployment on the STM32 Blue Pill (STM32F103) microcontroller.

2. System Overview

The proposed system consists of an IMU sensor, an STM32 microcontroller, a battery with basic battery management circuitry, and a LoRa communication module. The IMU captures accelerometer and gyroscope data, which is processed locally on the microcontroller. Machine learning inference is performed entirely on-device without relying on cloud computation.

Due to hardware unavailability, the system is demonstrated using an offline IMU dataset. However, the signal processing pipeline, model selection, and power considerations are fully compatible with STM32 deployment.

3. Dataset Description

The UCI Human Activity Recognition (HAR) dataset is used for this project. The dataset contains real IMU data collected from a waist-mounted smartphone.

- Sensors: Triaxial accelerometer and gyroscope
- Sampling frequency: 50 Hz
- Activities: Walking, Sitting, Standing, Lying, Walking Upstairs, Walking Downstairs

Only accelerometer signals are used to simulate a minimal wearable sensor setup suitable for embedded systems.

4. Signal Processing Pipeline

4.1 IMU Sampling Strategy

The IMU data is sampled at 50 Hz, which is sufficient to capture human motion patterns while keeping computational requirements low. For simplicity and embedded feasibility, only one accelerometer axis is selected.

4.2 Windowing

Continuous IMU data is divided into fixed-length overlapping windows to enable feature extraction and classification.

- Window size: 128 samples (≈ 2.56 seconds)
- Overlap: 50% (64 samples)

This window length captures complete gait cycles while maintaining real-time processing capability.

4.3 Feature Extraction

From each window, simple time-domain features are extracted:

- Mean
- Standard Deviation

These features are computationally inexpensive and effective for distinguishing motion intensity, making them suitable for STM32-based TinyML applications.

5. Machine Learning Model

5.1 Model Selection

A Decision Tree classifier is chosen for this project due to the following reasons:

- Low memory footprint

- Fast inference time
- Minimal computational complexity
- Easy conversion to embedded C code

These properties make Decision Trees well-suited for TinyML deployment on resource-constrained microcontrollers.

5.2 Training and Evaluation

The model is trained offline using extracted features and corresponding activity labels. The trained model achieves an accuracy of approximately **46%**. The moderate accuracy is expected due to the use of a single sensor axis and limited feature set.

The primary objective of the project is embedded feasibility rather than maximum classification accuracy.

6. STM32 Embedded Constraints

The STM32F103 microcontroller has limited RAM and flash memory. The chosen feature set and Decision Tree model ensure:

- Minimal memory usage
- Deterministic and fast inference
- Compatibility with real-time execution

Floating-point operations can be replaced with fixed-point arithmetic if required to further optimize performance.

7. Power and Battery Management

The system is designed to operate on a battery-powered supply.

- Battery type: 3.7V Li-Po
- Protection and charging: TP4056 module
- Protection features:
    - Over-voltage protection
    - Under-voltage protection
    - Over-current protection

STM32 low-power sleep modes can be used between inference windows to reduce power consumption and extend battery life.

8. Communication

LoRa communication is included for transmitting classification results to an external receiver. Raw sensor data is not transmitted, reducing communication power consumption and preserving privacy.

9. Results and Discussion

The implemented system successfully demonstrates a complete TinyML pipeline from IMU signal processing to motion classification. While accuracy is moderate, the system meets all embedded constraints and demonstrates feasibility for real-time wearable deployment.

Limitations include the use of a single accelerometer axis and a minimal feature set. These limitations can be addressed in future work by incorporating additional sensor axes and features.

10. Conclusion

This project demonstrates the design and implementation of a wearable gait and posture recognition system using TinyML principles. The system performs on-device inference under strict memory and power constraints and is suitable for deployment on the STM32 Blue Pill microcontroller. The project successfully meets all assignment requirements and provides a strong foundation for real hardware implementation in the future.

11. Future Enhancements

- Use all accelerometer and gyroscope axes
- Add frequency-domain features
- Implement fixed-point inference
- Deploy and test on actual STM32 hardware