Cyclify

C4: Rita Paixao, Zoe White, Carolyn Yang
18-500 Capstone Design, Spring 2025
Electrical and Computer Engineering Department
Carnegie Mellon University

Product Pitch

Cyclify is an **unobtrusive handlebar-and-saddle system** that embeds 13 FSR sensors to learn your unique pressure distribution on the bike. After a **two-minute, app-triggered calibration** that builds personalized baselines, it continuously applies an intensity-based detection algorithm, **achieving 84.38 % overall accuracy** with a false-alarm rate under 3.45 %. When your posture drifts, Cyclify delivers a Bluetooth alert to **your phone** in just 210 ms on average, giving you **immediate feedback** without disrupting your ride.

Housed in a **low-profile, bike-compatible enclosure,** Cyclify secures all electronics and power management circuitry for **six hours of continuous operation**. Built-in LEDs guide you through setup and calibration, and installation requires no tools or permanent modifications.

Cyclify can detect posture deviations in **210 ms** on average.
Cyclify can run continuously for **6 hours** on a single charge.
Cyclify can calibrate to your personal baseline in only **2 minutes**.
Cyclify achieves **84.38 % overall accuracy** with a **false-alarm rate under 3.45 %**.

System Architecture

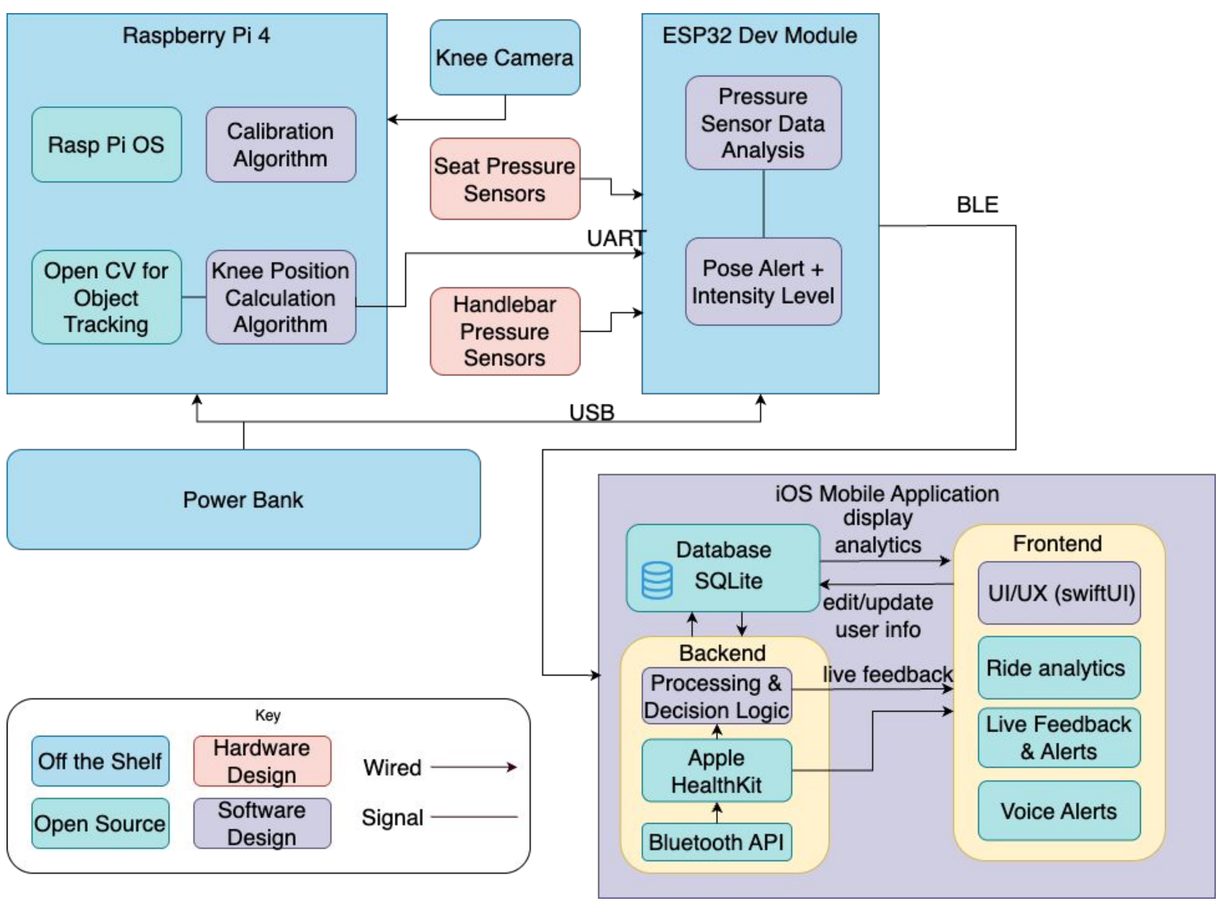


Figure 1. Cyclify System Architecture - Includes on-bike pressure sensors, embedded posture-detection processing, and wireless mobile alert delivery

Conclusions & Additional Information

What Worked
Cyclify provides cyclists with a durable, unobtrusive platform for posture feedback and injury prevention: real time force sensing via FSRs, pose-risk detection using thresholding and matching algorithms, wireless ESP32 to PI communication, data logging/analysis, and app based feedback. In addition, a knee tracking algorithm running on a Raspberry Pi 4 offers further real time knee position, enhancing the system's sensitivity to subtle rider imbalances. The companion app provides intuitive data visualization, providing riders with clear feedback during and after rides.

Lessons Learned
(1) Ordering parts as early as possible prevents any delays in design/development
(2) 3D printing can be unreliable and much more time consuming than expectation
(3) Everything we thought initially would be easy turned out to be more difficult/time consuming than expected

https://course.ece.cmu.edu/~ece500/projects/s25-teamc4/

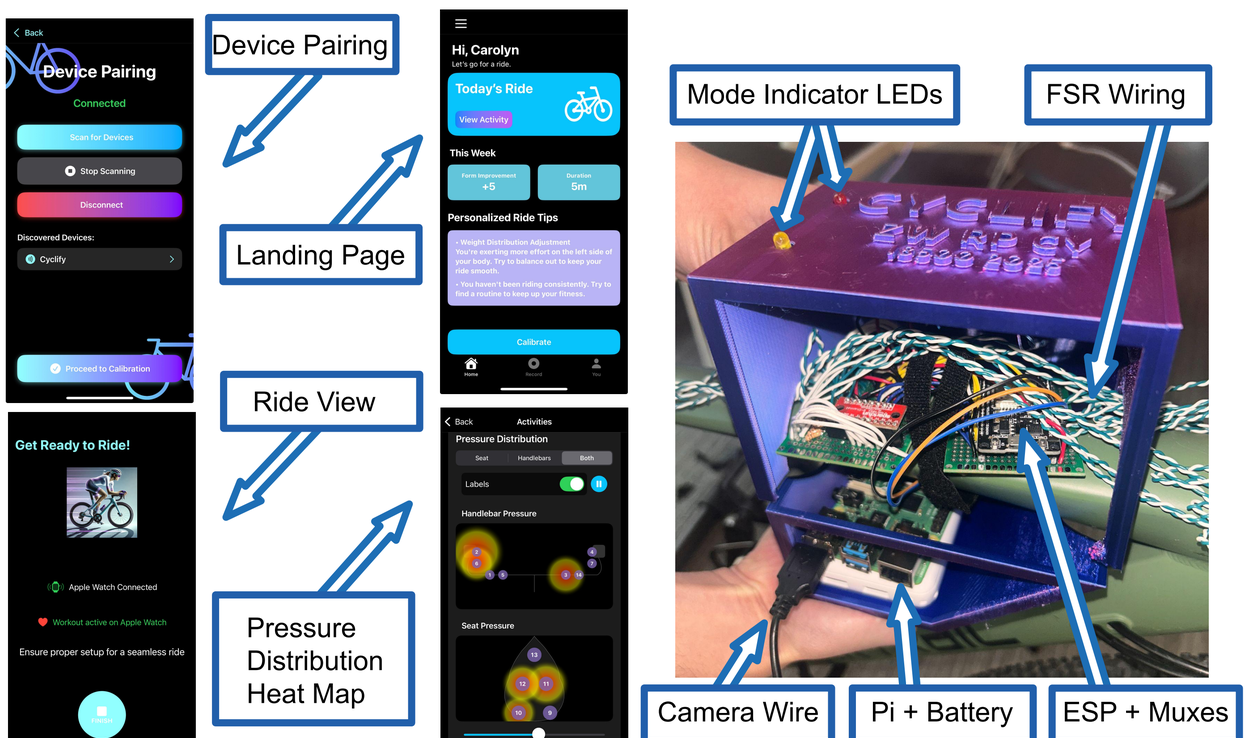
Room for Improvement
Develop custom PCB to consolidate hardware
Design snap to fit device housing for easy assembly
Create FSRs slip on mounts for saddle/handlebars
Integrating AI modules for overuse/exercise pattern recognition
Provide personalized recognition for targeted workouts

System Description

Posture Detection System
Cyclify merges 13 FSR readings, routed through dual 8-to-1 multiplexers into an ESP32, with live knee-position data from a Pi camera. The camera tracks lateral displacement, streams coordinates over UART, and the ESP32 applies a two-stage pipeline (hard override + weighted MAE scoring) to compute a unified posture metric every cycle.

User Interface
The BLE-linked app guides you through a two-minute calibration, lets you fine-tune alert thresholds, and instantly notifies your phone or watch when form drifts. It logs every event in SQLite for ride analytics, shows real-time and post-ride dashboards, and offers optionai am I voice cues and training insights.

Enclosure & Electronics
All hardware, custom protoboard, ESP32, multiplexers, RPi 4, LEDs, and 5 V battery, sits in a 3D-printed housing that can secure to most bike frames. It provides LED status indicators and over six hours of continuous operation per charge.



System Evaluation

Use-Case & Design Requirements :

Metric	Target	Actual
Battery Life	≥ 2 hours	≥ 5 hours
Overall System Accuracy	≥ 80%	~ 84.38%
False Alarm Rate	≤ 5%	≤ 3.45%
Device Latency	≤ 500ms	≤ 210 ms
Wireless Connectivity	≥ 5m	≥ 10m

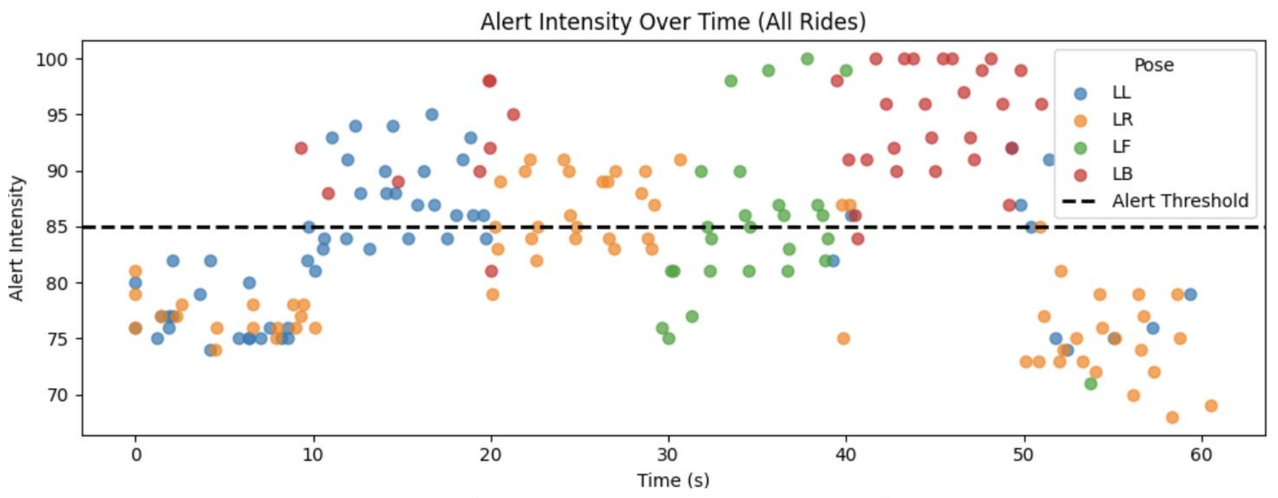


Figure 2. Alert Intensity vs. Time (All Rides)

Figure 2 shows our 85-point threshold cleanly flags real posture shifts, each burst of points above the line follows with the rider's 10-second planned bad pose windows during the test rides, while leaving normal riding variation below the alert level.

Delivering that level of precision meant balancing expansive sensing against circuitry and power constraints. Adding 13 FSRs plus a knee-position camera boosts fidelity but increases wiring and current draw, so we multiplex inputs, streamline our override + MAE algorithm and optimize JSON payloads for efficient BLE transmission, and leverage the ESP32's low-power modes to preserve six-hour battery life.