Team E7: Body Buddy

Nick Lee, Sojeong Lee, Max Lutwak, Jacob Hoffman

Application Area

- Problem
  - Falls can cause serious injuries for elders
  - Fear of falling can also limit their activities / social engagements

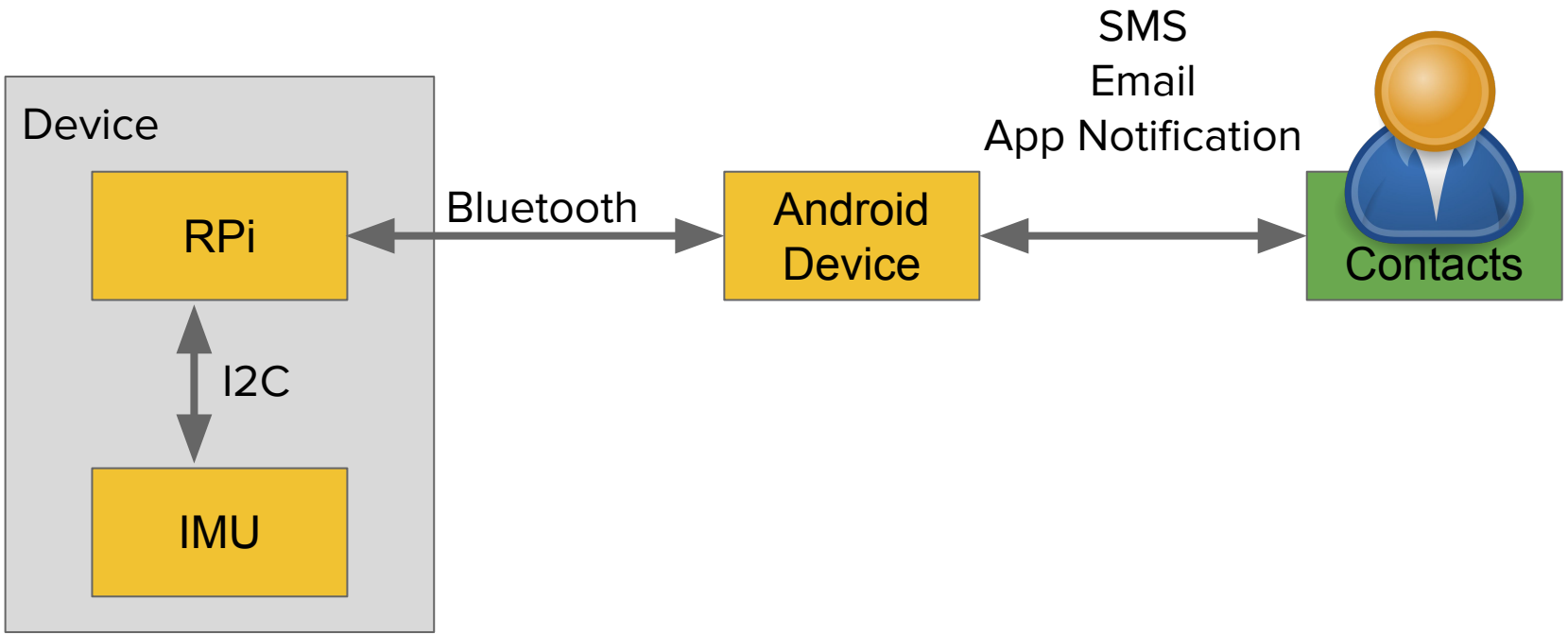
- How can we promptly handle the emergency situations caused by falls?

  ➜ An attachable device connected to a mobile app that detects a fall and sends alerts to the first responders

Solution Approach

- Data collection
  - 3-axis accelerometer
- Fall detection
  - Train two ML approaches on the data (SVM, RNN)
- Alert system
  - Mobile app sending alerts to the contacts (first responders)
- Device design
  - Minimize size and weight, maximize battery life

Block Diagram



Implementation Plan - Hardware

- Main platform: Raspberry Pi Zero W
  - Low power
  - Bluetooth Low Energy (BLE) & I2C
  - Full OS, so we can choose to do ML locally
- IMU: Sunfounder ADXL345 board
  - 3-axis accelerometer
  - >100 samples/sec over I2C
  - Small form factor, low power draw (<5mA)
- Power Supply: Attom Tech 3000mAh smartphone charger
  - Similar dimensions to Pi Zero case
  - Expected power draw is <200mA, should guarantee >10hrs
  - Lightweight (2.2oz)

Implementation Plan - ML

- Train two machine learning systems on the data
  - SVM
  - RNN
- Compare and contrast the performance tradeoffs
  - Bias, Variance, Accuracy, Loss
- Use a sliding window of 10 seconds interval to run the algorithm
  - If needed, can improve efficiency by running only on a big change in data

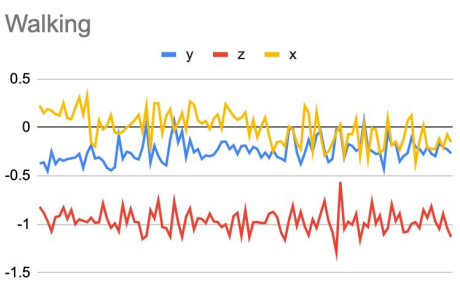
Implementation Plan - Data

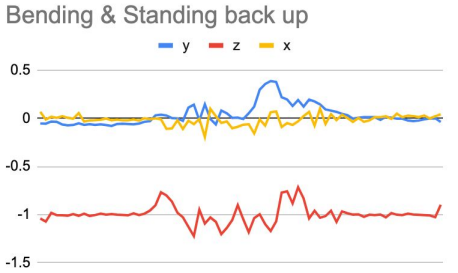
- Collect a dataset of simulated falls / normal activities
  - Take some falls for the team
  - Get a dummy and attach our hardware
- Manually label our data
  - Use a tool such as TRAINSET
- Segment the data
  - Allows an SVM to classify the falls.
- Apply a Kalman Filter
  - Smooths the data out
  - More accurately interpreted by our ML algorithms.

Falls	Normal Activities
Falling forward / backward	Walking
Falling sideways	Running
Falling from stairs	Jumping
Falling on an incline	Lying down
Falling on a decline	Sitting / Bending Down

Sample Data

- 3-axis acceleration
  - Can add more data (gyroscope, magnetometer) if desired accuracy is not achieved
- Collected using iPhone accelerometer (50Hz data rate)
  - Will get more accurate data on Pi
- Can spot differences in fall and non-fall graphs





Implementation Plan - Mobile App

- Alert system
  - Allow 2 minutes for users to cancel the alarm
  - Send automatically alerts to saved contacts after 2 minutes
- Leverage Android Studio to make a mobile application with major features
  - Bluetooth API - connection with RPi
  - Send location in a human-readable format (Location API)
  - Contacts Provider - manages the contact information data
  - SmsManager - sending SMS messages

Metrics and Validation

- **Hardware**
  - Battery Life (>10h)
    - Leave system running until we stop transmitting data
    - Mostly just a function of battery choice
  - Weight (<10oz.)
    - A scale
- **Mobile App**
  - Connections (RPi <-> App / App <-> Contacts)
    - Send dummy data to measure latencies for messaging services
  - Location
    - Determine the correct location 95%+ of the time
  - Front-End UI
    - User testing

Metrics and Validation

- **Fall Detection**
  - Clear falls / normal activities categories to ease testing
  - Calculate accuracy of the algorithm (>90%) test data from each category
  - Risk Factors

$$\frac{True\ Positive\ +\ True\ Negative}{True\ Positive\ +\ True\ Negative\ +\ False\ Positive\ +\ False\ Negative}$$

    - Low accuracy of the algorithm
      - Two approaches (SVM, RNN)
      - Try training the model with different features
        - Tuple of x, y, z accelerations
        - Total magnitude of accelerations
        - Angle of acceleration
    - Discrepancy between real-world & test data
      - Use a dummy for collecting large set of fall data, but also collect actual fall data using a gym mat
      - Collect data from people with different weights / heights

Project Management