In-Ear Activity Monitoring: Using Apple AirPods for Physical Activity Recognition

Dylan Edwards
dylanme2@illinois.edu

Kyler Yu
kyleryu2@illinois.edu

ABSTRACT

This project explores an innovative application of Apple AirPods for physical activity recognition, leveraging their integrated inertial sensors and accelerometers. Current wearable technologies, such as wrist-worn devices, face challenges related to cost and comfort. By shifting focus to headphones, which are already widely adopted, this study introduces a novel method for activity detection using the CMHeadphoneMotionManager API.

The approach employs a four-step pipeline: data collection, pre-processing/de-noising, activity classification, and feature extraction for user feedback. Motion data is retrieved from the AirPods and processed to derive acceleration and orientation information in a global reference frame. The data is then smoothed with a Savitzky-Golay filter for classification.

A CNN-LSTM model architecture, combining convolutional layers for spatial feature extraction and LSTM layers for temporal pattern recognition, performs real-time classification of activities such as walking or running. Features like step count, intensity, and distance are subsequently extracted to enhance user feedback. Data is processed in five-second intervals, ensuring seamless real-time analysis through a dual-buffer system that prevents race conditions.

The user interface complements the backend functionality, providing clear visualization of activity data, historical trends, and real-time feedback. Initial evaluations demonstrate the system's effectiveness by achieving high accuracy.

ACM Reference Format:
Dylan Edwards and Kyler Yu. 2024. In-Ear Activity Monitoring: Using Apple AirPods for Physical Activity Recognition. In . ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The advancement of technology has brought about significant transformations in both the healthcare and sports industries, largely driven by the introduction of wearable technologies. These innovations, particularly in the areas of activity tracking and overall health monitoring, have played a pivotal role in enhancing personal fitness and well-being. Among the many devices available, wrist wearables such as the Fitbit and Apple Watch have been at the forefront, providing users with a convenient and efficient means to track their physical activities and health metrics. Despite their widespread adoption, concerns regarding the cost and comfort of these products have prompted a closer look at alternative solutions. In particular, the question arises: How can physical activities continue to be effectively tracked without the reliance on traditional wrist-worn devices like watches?

Conference'17, July 2017, Washington, DC, USA
2024. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

A promising yet under-explored form of wearable technology lies in the realm of headphones. This category of devices has seen a surge in popularity, owing to their accessibility, convenience, and the recent advancements in technology, such as the integration of inertial sensors and accelerometers. By harnessing the power of these sensors in Apple AirPods, our project aims to develop a reliable and innovative method for physical activity recognition that goes beyond the limitations of conventional wrist-based devices.

The proposed solution consists of four key components: data collection, data cleaning and pre-processing, activity classification, and feature extraction with user feedback. These components are seamlessly integrated into a mobile application that not only detects physical activities but also provides real-time feedback to the user, helping them monitor their progress and improve their performance. As illustrated in Figure 1, these stages are designed to work together to offer a comprehensive and user-friendly experience.

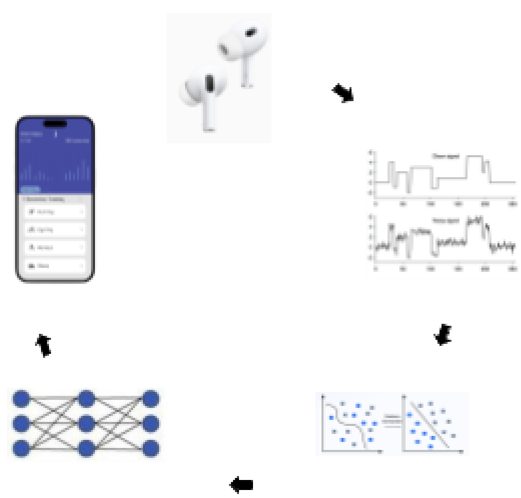

Figure 1: Application workflow

For the application to function accurately and effectively, it requires Apple AirPods to be connected to the device and worn in-ear, ensuring optimal sensor readings. Each of the four components plays a critical role in the success of the project, and in the following section, we will explore each component in greater detail, providing a deeper understanding of how they contribute to the overall functionality and effectiveness of the solution.

2 PROPOSED APPROACH

Our proposed approach leverages the capability to collect and analyze motion data directly from Apple AirPods. This is achieved through the use of Apple's CMHeadphoneMotionManager API, a library specifically designed for accessing motion-related data from headphones. The API provides access to accelerometer data, including both user acceleration and gravitational acceleration, as well as gyroscopic movement data collected via the Inertial Measurement Unit (IMU) embedded in the AirPods.

Once the motion data is retrieved from the headphones, it is passed into a pre-processing function designed to acquire the the acceleration into the x, y, and z axis in terms of the global coordinate system. Next, the data is then de-noised by utilizing a Savitzky-Golay filter to smooth out the spikes in data for easier pattern detection.

The cleaned data is then fed into a CNN-LSTM model, which is specifically designed to extract meaningful features while retaining important sequential patterns in the data. Initially, the data flows through convolutional layers that detect overarching trends and patterns. From there, it passes through LSTM layers dedicated to more detailed analysis and classification, where each subsequent layer hones in on increasingly specific aspects of the data.

After the data is classified, further analysis will be conducted on it depending on the type of activity it is. For example, step detection will be performed if the data is classified as walking or running.

Finally, as the model runs in real-time, it predicts and relays the identified activity and analysis directly to the user, providing immediate feedback based on the live data collected from the AirPods. This streamlined approach ensures accurate, context-aware activity recognition and enhances the user's overall experience by delivering predictions dynamically.

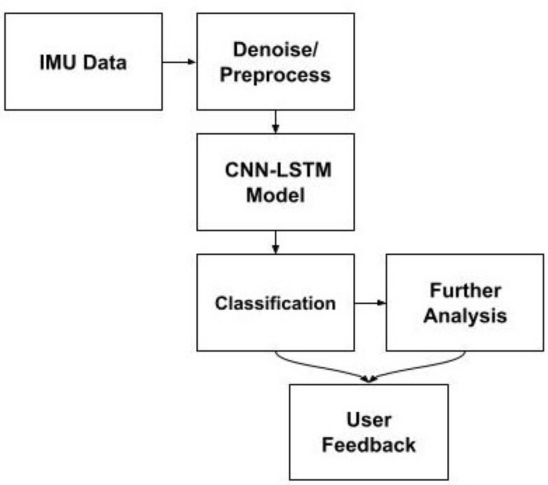

Figure 2: Data workflow

3 IMPLEMENTATION

Through the CMHeadphoneMotionManager API a multitude of data categories were available. These include: Quaternion X/Y/Z/W, Pitch/Roll/Yaw, Gravitational Acceleration X/Y/Z, Rotation Acceleration X/Y/Z, User Acceleration X/Y/Z, and Rotation Matrix. The data currently being processed by the application is the Timestamps, Rotation Matrix (Quaternion X/Y/Z/W), and User Acceleration in the X/Y/Z.

The data is then pre-processed to find the absolute acceleration of the headphones in the X/Y/Z-directions. This is done by taking the quaternion rotation matrix and multiplying it with the user accelerations to get the actual acceleration in the X/Y/Z directions with respect to the global reference frame.

The quaternion rotation matrix is derived from the orientation quaternion, which encodes the rotation from the local device frame to the global frame. The quaternion effectively expands Euler's Equation for 2-dimensional rotations into the third dimension. To do this, the quaternion represents a rotation using four components, $q = w + xi + yj + zk$, where $w = cos(\theta/2)$ (scalar part, linked to the angle of rotation), $x, y, z = sin(\theta/2) \times (v_x, v_y, v_z)$ (vector part, linked to the axis of rotation), θ is the rotation angle, and (v_x, v_y, v_z) is the unit vector defining the rotation axis. For example, a rotation of 90 degrees around the z-axis: $q = cos(45\circ) + sin(45\circ) \times (0i + 0j + 1k)$ results in the quaternion $q = 0.707 + 0.707k$.

The quaternion rotation matrix R is then used to transform the accelerometers' local frame data into the global frame by applying matrix multiplication to the raw acceleration vector. This method ensures that the acceleration data accounts for both the device's orientation and the movement of the user, providing a more accurate representation of real-world forces acting on the device. Next, the global frame data is aligned with the initial forward direction of the AirPods. To calculate the initial forward direction (AirPods' yaw in the world frame), we calculate $initYaw = arctan2(initForwardX, initForwardY)$, where $initForwardX$ is element 11 in the quaternion rotation matrix and $initForwardY$ is element 12. Next, we align the global frame data to the initial forward direction through the following computations: $forwardAccel = worldX \times cos(initYaw) + worldY \times sin(initYaw)$, $sideAccel = -worldX \times sin(initYaw) + worldY \times cos(initYaw)$, and $downAccel = worldZ$.

While a more simplistic method using Euler angles (pitch, roll, and yaw) could have been used to convert the local frame data into the global frame, using it could lead to several errors. One such error is gimbal lock, which is when two of the three rotation axis align resulting in a loss of one of the degrees of freedom. For example, with Euler angles, if the pitch reaches 90 degrees, the yaw and roll axes would align, leaving the system unable to distinguish between certain rotations. On the other hand, quaternions prevent gimbal lock by representing rotations as a four dimensional vector, as stated explained previously. The four components encode the rotation without explicitly depending on intermediate angles like Euler angles. Furthermore, rather than representing a rotation as three sequential transformations, quaternions represent rotation as a single transformation.

$$M_R = \begin{pmatrix} 1-2y^2-2z^2 & 2xy-2sz & 2xz+2sy & 0 \\ 2xy+2sz & 1-2x^2-2z^2 & 2yz-2sx & 0 \\ 2xz-2sy & 2yz+2sx & 1-2x^2-2y^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 3: Quaternion Rotation Matrix Calculation (s = w)

Once the accelerometer data is rotated into the global frame, the data is then denoised by a Savitzky-Golay low-pass filter to make the data easier to interpret for the model. The filter addresses this by smoothing the data through a process that involves fitting a polynomial (typically quadratic or cubic) to a sliding window of data points. The center of the window is then replaced with the value predicted by this polynomial. The reason for selecting this method is that it helps reduce rapid, small fluctuations (high-frequency noise) while preserving the overall trend and important features of the data to make classification easier. Finally, additional normalization steps are applied to refine the measurements, ensuring they are accurate and ready for further analysis.

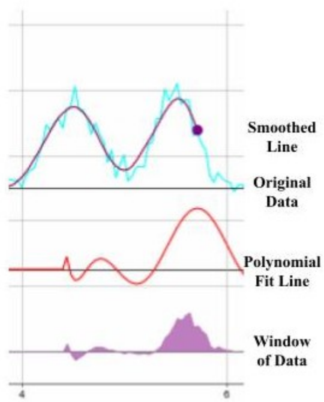

Figure 4: Savitzky-Golay Low-pass Filter Visualization

The preprocessed data is then split into five-second chunks to allow for real-time analysis and passed as input to a CNN-LSTM model. Long Short-Term Memory (LSTM) models, a type of advanced recurrent neural network (RNN), are specifically engineered to process sequences of data, making them ideally suited for time-series analysis where past information is key to predicting future events. Unlike standard RNNs that struggle with long-range dependencies due to issues like vanishing gradients, LSTMs effectively retain information over extended periods thanks to their unique internal structure. This structure consists of units called cells, which contain mechanisms called gates that regulate the flow of information. These gates—input, forget, and output—control the entry of new data into the cell, the retention of existing data, and the output of data from the cell, respectively.

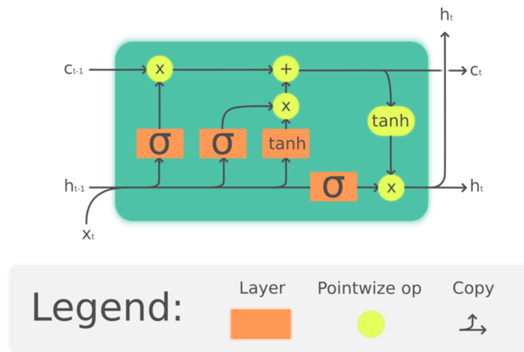

Figure 5: LSTM Model Visualization

To enhance the predictive accuracy of LSTM (Long Short-Term Memory) models, we integrated Convolutional Neural Network (CNN) layers into the architecture. This integration leverages CNNs' ability to process and recognize spatial patterns, augmenting the model's capacity to handle temporal sequences with complex spatial hierarchies.

CNNs excel at extracting high-level features from spatial input through convolutions, which apply filters to identify significant features independent of their specific location. By incorporating CNN layers, the model preprocesses input sequences, breaking them down into meaningful patterns before feeding them into the LSTM layers. This preprocessing enables LSTMs to focus on temporal dynamics rather than decomposing raw data into features.

In the context of physical activity classification using accelerometers, spatial hierarchies may refer to patterns like repetitive motion cycles within broader activity trends. For example, step patterns in walking or running may exist within overall activity sequences. CNN layers can capture these patterns as spatial features, which are then analyzed through the temporal lens of LSTMs.

This setup benefits the LSTM layers by providing feature-enriched inputs, improving their ability to understand complex temporal dependencies. Additionally, CNN layers reduce the dimensionality and complexity of input data, lightening the computational load on LSTMs. This not only accelerates the training process but also enhances convergence, making the approach efficient and practical for real-world applications.

In essence, integrating CNN layers with LSTM models creates a powerful architecture for handling data with intricate spatial-temporal characteristics, improving both accuracy and computational efficiency in tasks like physical activity classification.

Once a specific activity is classified by this model, certain features are then extracted from the data depending on the type of activity. For example, if the activity is running or walking, features such as step count, intensity, time, and distance could be calculated. These features are important in providing valuable feedback and analysis for the user.

To implement real-time analysis, we collect data in five-second chunks and send these small chunks to the analysis script. In order to both ensure that there are no gaps in data and avoid race conditions (which would occur if the data buffer is being read by the analysis program and written to at the same time), two buffers are used: one for writing and one for reading. After a five-second period is finished, the writing buffer is swapped with the reading buffer, and the new writing buffer is cleared for the next five-second period. Then, while the application continues to write sensor data to the write buffer, the analysis program reads from the read buffer.

Finally, we developed an elegant user interface that is able to display all of the analysis and data from the application. In order to highlight specific activities we developed individual pages for users to click into, where they can find extra information that pertains to that activity such as historical analysis for distance with regard to running. The application reveals basic data to the user via the front page that includes overall steps and calories burned for the current day.

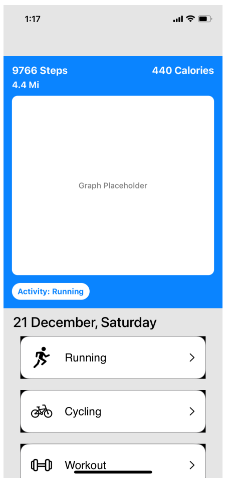

Figure 6: Home Page Visual

4 EVALUATION

In order to evaluate our system's performance, we considered the accuracy of the classification model. This was done by simply comparing the actual classification of the data collected with the output of the model. These datasets were tested using an 80-20 split, meaning 80 percent of the data was used for training and 20 percent for testing. As a result we were able to achieve a training accuracy of 93.2 percent and a testing accuracy of 86.7.

5 CONCLUSION

We have developed a sleek and robust user interface, paired with an effective real-time analysis script that provides immediate feedback to the user. Our project successfully processes raw data from Apple AirPods and translate it into specific activities. We achieved this by employing a quaternion rotation matrix to obtain reliable acceleration data, effectively minimizing errors that may arise when using either using the raw data or a rotation matrix based on gyroscope data (roll, pitch, yaw). To further refine this data, we applied a Savitzky-Golay low-pass filter, reducing noise while preserving key information. This cleaned data was then segmented into 5-second chunks and input into a CNN-LSTM model for real-time classification, ultimately leading to classification and activity-related features being given as user feedback.

The project has significant potential for expansion, as the current portfolio of activities is limited. To enhance the project, we aim to develop profiles for additional activities, such as various workout routines, cycling, and sleeping. By acquiring more datasets and analyzing the differences between these activities, we can improve the performance of the classification model. Additionally, we plan to increase data flow efficiency by integrating classification and feature extraction within the same model. Lastly, we aim to implement an inward-facing microphone to enhance health tracking capabilities, allowing us to monitor factors like eating habits (chewing and drinking), sleep patterns (teeth grinding), and heart rate. These improvements will contribute to better tracking of activity intensity and overall health.

GitLab Link: https://gitlab.engr.illinois.edu/cs437fa24group18/airpods-activity-tracking.git