HARmony

Jeonghoon Park
UNIST
hoonably@unist.ac.kr

1 Introduction

Human Activity Recognition (HAR) identifies a user's physical activity from sensor data collected by mobile or wearable devices. In smartphone-based HAR, the accelerometer and gyroscope are particularly useful because they capture complementary aspects of motion, namely linear acceleration and rotational movement.

In this project, we developed a fully rule-based HAR system using only the built-in accelerometer and gyroscope. The system recognizes the five required activities—*Still*, *Walk*, *Running*, *Stairs Up*, and *Stairs Down*. In addition to the five required activities, the system also includes *Moonwalk* as an extra activity to test whether the rule-based pipeline can distinguish a more subtle gait-like motion from ordinary walking and stair-related movement.

To support the full pipeline, we also implemented an iOS application for labeled data collection and real-time activity recognition, using the same features and decision rules in both offline analysis and online deployment. The source code is available at: `https://github.com/hoonably/HARmony`

2 Data Collection

2.1 Experimental Setup

Sensor data were collected using the smartphone's built-in accelerometer and gyroscope at 50 Hz. All activities were recorded under the same handheld condition: the phone was held in the participant's right hand, and the participant continuously viewed the screen while performing the activity. The device posture was kept relatively stable to mimic natural smartphone use during walking, which is often associated with a forward/downward viewing posture.

We collected the five required activities—*Still*, *Walk*, *Running*, *Stairs Up*, and *Stairs Down*—as well as one additional activity, *Moonwalk*, using the same protocol. For feature analysis, rule design, and threshold tuning, each activity was recorded for 10 trials of 5 s each. For offline evaluation, a separate test set was collected independently. For *Still*, *Walk*, *Running*, and *Moonwalk*, about 30 s of continuous test data were recorded per activity, whereas *Stairs Up* and *Stairs Down* were evaluated using six 5 s trials concatenated together because a single staircase segment was not long enough for a continuous 30 s recording.

Walking was performed at a normal pace along a straight path. Running was performed on a flat outdoor path at the fastest pace that still allowed the participant to view the screen. Stair data were collected on the stairs of UNIST Building 106, where each step had a height of approximately 18 cm. Since one ascent or descent took about 6.5 s, 5 s trials were used to capture only the continuous stair-motion segment without including standing still before or after the movement.

2.2 Recording Protocol

To reduce artifacts caused by touching the screen, recording started 1 s after the `Start` button was pressed and stopped automatically after the preset duration. This protocol was applied consistently so that each 5 s trial contained a relatively pure motion segment with minimal screen-interaction noise, which was particularly important for stair-related activities.

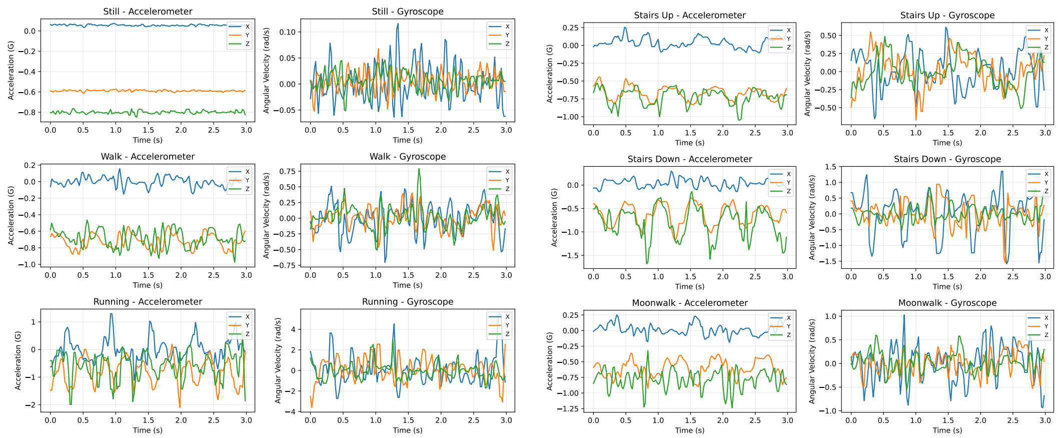

Figure 1: Representative raw sensor signals for the six target activities. The left panel shows *Still*, *Walk*, and *Running*: *Still* has near-flat accelerometer and gyroscope signals, *Walk* shows moderate periodic motion, and *Running* shows larger oscillations and stronger rotational movement. The right panel shows *Stairs Up*, *Stairs Down*, and *Moonwalk*, which all have locomotion-like patterns but differ in impact level, periodic structure, and directional coupling across axes.

3 Feature Analysis

3.1 Observed Sensor Patterns

Before defining the final rules, we inspected representative accelerometer and gyroscope signals for each activity, as shown in Fig. 1. *Still* showed near-constant signals, *Walk* showed moderate periodic motion, and *Running* showed larger fluctuations with stronger rotational movement. *Stairs Up*, *Stairs Down*, and *Moonwalk* also appeared locomotion-like, but differed in impact level, periodic structure, and directional coupling. These observations motivated the use of time-domain, frequency-domain, and inter-axis correlation features.

3.2 Selected Features

We first converted the three-axis sensor signals into acceleration and gyroscope magnitudes:

$$a_{\text{mag}} = \sqrt{a_x^2 + a_y^2 + a_z^2}, \qquad g_{\text{mag}} = \sqrt{g_x^2 + g_y^2 + g_z^2}.$$

For each sliding window, we extracted seven features: the standard deviation of acceleration magnitude, the mean of gyroscope magnitude, the spectral energy and spectral entropy of detrended acceleration magnitude, the ratio between the second-largest and largest frequency peaks, and two inter-axis correlations, namely those between a_y and a_z and between a_x and a_y.

These features were selected because they captured complementary aspects of motion: intensity, rotation, frequency structure, and axis coupling.

3.3 Interpretation of the Selected Features

Acceleration magnitude standard deviation and gyroscope magnitude mean were the most useful for separating low-motion and high-motion activities. *Still* consistently produced very small values, whereas *Running* produced much larger values due to stronger impact and rotation.

Magnitude alone, however, was not sufficient to separate *Walk* from stair-related activities. We therefore additionally used spectral and correlation-based features. Spectral energy measured the strength of periodic motion, while spectral entropy and peak-ratio described how regular or complex the frequency structure was. Among stair-related activities, *Stairs Up* tended to show clearer positive correlation between the y and z acceleration axes, whereas *Stairs Down* generally showed stronger impact and higher spectral energy.

The additional activity *Moonwalk* also required these higher-level cues. Although it resembled walking at a coarse level, it showed a more irregular frequency structure and different directional relationships between axes. Overall, the selected feature set was sufficient to support an interpretable rule-based HAR system without using a learned model.

4 Rule-Based HAR System Design and Implementation

4.1 Processing Pipeline

The final system was implemented as an iOS application with two main tabs: data collection and real-time recognition. The app reads accelerometer and gyroscope data at 50 Hz, maintains a fixed-length sliding window, extracts the selected features, and applies the rule-based classifier to each window. The same feature set was used in both offline analysis and online recognition to keep the deployed system consistent with the analysis pipeline.

In recognition mode, the app uses a 1.8 s window and updates the prediction periodically. To reduce rapid label flickering, the displayed result is smoothed using a majority vote over recent predictions.

4.2 Feature Computation and Classification Rules

For each sliding window, the system computes acceleration and gyroscope magnitudes and extracts seven lightweight features (Section 3), enabling real-time on-device processing.

The rule-based classifier is summarized in Algorithm 1 and follows a sequential decision structure. First, extreme cases are identified: *Still* corresponds to very low motion intensity and rotation ($\sigma_a < 0.08$, $\mu_g < 0.25$), while *Running* exhibits consistently large values ($\sigma_a > 0.38$, $\mu_g > 1.15$).

Next, stair-related activities are detected using spectral energy and inter-axis correlation. *Stairs Up* is characterized by moderate energy ($30 < E < 150$) and strong positive coupling between the y and z axes ($\rho_{yz} > 0.35$), reflecting periodic upward motion. In contrast, *Stairs Down* typically produces higher impact and energy ($E > 220$) with even stronger correlation ($\rho_{yz} > 0.45$).

For the remaining non-extreme cases, the classifier next checks moonwalk and stair-like conditions using higher-level frequency and correlation cues. Moonwalk is identified by high spectral energy ($E \geq 115$), irregular frequency structure (large R_2), and high entropy ($H > 1.60$), together with weak or negative axis correlation. Stairs Up is then detected using moderate energy, regular frequency structure, and positive y–z axis coupling, and all remaining cases are classified as *Walk*.

Algorithm 1 Rule-based activity classification for one sliding window

Require: $\sigma_a, \mu_g, E, H, R_2, \rho_{yz}, \rho_{xy}$
walkLike $\leftarrow (0.08 < \sigma_a < 0.35) \wedge (20 < E < 380) \wedge (\mu_g < 1.6)$
stairsUp $\leftarrow (25 < E < 170) \wedge (R_2 < 0.85) \wedge (\rho_{yz} > 0.05)$
moonwalk $\leftarrow (E \geq 115) \wedge (R_2 > 0.30) \wedge (H > 1.60)$
    $\wedge\big((\rho_{yz} < 0.10) \vee (\rho_{xy} < -0.05)\big)$
if $\sigma_a < 0.08 \wedge \mu_g < 0.25$ **then**
    return STILL
else if $\sigma_a > 0.38 \wedge \mu_g > 1.15$ **then**
    return RUNNING
else if $\rho_{yz} > 0.35 \wedge 30 < E < 150$ **then**
    return STAIRSUP
else if $\rho_{yz} > 0.45 \wedge E > 220$ **then**
    return STAIRSDOWN
else if *moonwalk* $\wedge$ (*walkLike* $\vee \neg$*stairsUp*) **then**
    return MOONWALK
else if *stairsUp* **then**
    return STAIRSUP
else
    return WALK
end if

4.3 Window Size Selection

Since all features were computed over fixed-length sliding windows, the window size directly affected both feature stability and real-time responsiveness. We therefore evaluated multiple window sizes using the same rule-based pipeline, and the corresponding accuracy trend is shown in Fig. 2. For offline evaluation, windows were slid with a stride of one sample, corresponding to 0.02 s at 50 Hz.

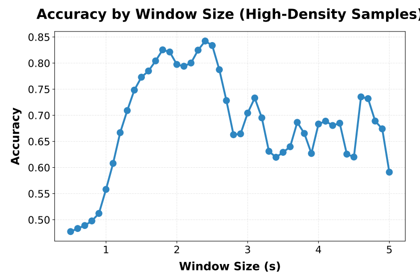

Figure 2: Recognition accuracy versus window size.

The overall trend shows that performance improved substantially from short windows to medium-length windows and then decreased again for longer windows. The best offline accuracy was obtained at 2.4 s, while several other local peaks also appeared around 1.8 s, 3.1 s, and 3.7 s. In contrast, lower accuracies were observed around intermediate values such as 2.1 s, 2.8 s, 3.4 s, 3.9 s, and 4.5 s.

This pattern was closely related to the measured walking step interval, which was approximately 0.556 s. Window sizes such as 1.8 s and 2.4 s are close to integer multiples of this step period, so they tend to contain more consistent numbers of repeated gait events and produce more stable averaged features. By contrast, when a window includes an additional half-step or other partial gait segment, the averaged features can change more noticeably, making the activity patterns less consistent across windows and reducing classification accuracy. This helps explain why some intermediate window sizes performed worse than nearby peak values.

At the same time, performance dropped again for sufficiently long windows, likely because they contained more intra-activity variation and fewer valid windows from each 5 s trial.

Although 2.4 s yielded the best offline accuracy, the final mobile app used a 1.8 s window because it provided similar recognition performance with faster updates. Based on this trade-off, 1.8 s was selected as a practical compromise between accuracy and responsiveness.

5 Results and Demonstration of the Final App

Table 1: Per-class performance comparison for the 1.8 s app window and the best 2.4 s offline window.

Activity	1.8 s			2.4 s		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Still	1.00	1.00	1.00	1.00	1.00	1.00
Walk	0.71	0.58	0.64	0.59	0.66	0.63
Running	1.00	1.00	1.00	1.00	1.00	1.00
Stairs Up	0.56	0.99	0.72	0.69	0.80	0.74
Stairs Down	0.96	0.98	0.97	0.88	1.00	0.94
Moonwalk	0.99	0.40	0.57	0.99	0.60	0.74
Accuracy		0.8254			0.8423	

5.1 Quantitative Results

The quantitative results were computed on a separately collected self-recorded test set, rather than on the data used for threshold tuning. Across multiple fixed window sizes, medium-length windows performed substantially better than very short windows, and the best offline accuracy was obtained at 2.4 s. Table 1 compares the per-class precision, recall, and F1-score for the 1.8 s window used in the mobile app and the best-performing 2.4 s offline window.

Still and Running were recognized almost perfectly in both settings. The 1.8 s window achieved an overall accuracy of 0.8254, while the 2.4 s window achieved the best offline accuracy of 0.8423. Although the 2.4 s window slightly improved the overall class balance, the 1.8 s window still provided comparable performance and was selected for the final mobile app because it allowed faster and more responsive updates.

5.2 Error Analysis

Figure 3 shows that the main source of confusion in the final 1.8 s setting was between *Walk* and *Stairs Up*. This is expected because both are periodic locomotion activities with similar overall motion intensity under the handheld phone condition used in this study. In the 1.8 s window, *Stairs Up* showed very high recall but relatively low precision, indicating that many true stair-ascent windows were correctly detected, but a considerable number of walking windows were also absorbed into this class. This suggests that the current stair-related rules are sensitive to upward stair motion, but not yet selective enough to fully separate it from ordinary walking.

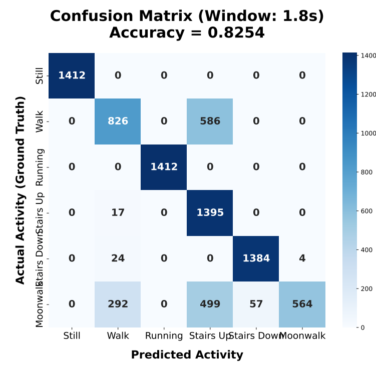

Figure 3: Confusion matrix for the 1.8 s window used in mobile app.

A second difficulty appeared in the *Moonwalk* class. Although *Moonwalk* achieved very high precision, its recall remained much lower in the 1.8 s window, meaning that moonwalk predictions were usually correct, but many actual moonwalk windows were still missed and instead classified as *Walk* or stair-related activities. This is reasonable because moonwalk shares a coarse gait-like structure with other locomotion classes while differing mainly in higher-level frequency irregularity and directional coupling. Overall, the results suggest that the added spectral and correlation-based features were useful for the extra activity, even though the class remained challenging.

5.3 Mobile App Interface and Demo Video

Figure 4 shows representative screenshots of the implemented iOS application. The data collection screen provides live accelerometer and gyroscope plots, recording duration control, activity label selection, recording controls, and CSV export. The recognition screen shows live sensor signals together with the activity currently predicted by the rule-based classifier in real time.

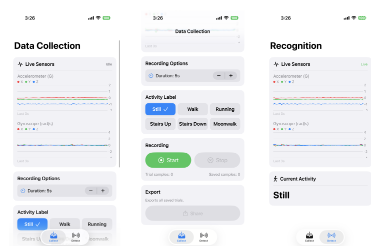

Figure 4: Screenshots of the implemented iOS application.

The demo video shows synchronized real-time recognition while the participant performs Still, Walk, Running, Stairs Up, and Stairs Down, with both the participant and app screen visible simultaneously.

6 Discussion and Conclusion

Overall, the proposed system worked well for activities with clearly distinct motion patterns, especially *Still* and *Running*. The most difficult case remained separating *Walk* from *Stairs Up*, and the additional *Moonwalk* class further highlighted the limitations of simple threshold-based rules for subtle gait-like motions. Even so, the selected handcrafted features provided an interpretable separation between the major activity types.

From a deployment perspective, the 1.8 s window provided a practical balance between recognition performance and responsiveness, whereas the 2.4 s window provided the best offline accuracy. These results suggest that the proposed rule-based HAR pipeline is effective for real-time smartphone sensing, while still leaving room for refinement in handling similar locomotion patterns.

This project implemented an end-to-end smartphone-based HAR system using only the accelerometer and gyroscope. The final system achieved reasonable offline performance on a self-collected test set and demonstrated synchronized real-time recognition in the implemented iOS application and demo video.

Although confusion remained between *Walk* and *Stairs Up*, the overall system successfully demonstrated the complete HAR pipeline required in the assignment. Future work could consider more diverse phone positions, more robust cross-session testing, and more adaptive rules for similar locomotion patterns.