

Research Article

A Smart Glove to Track Fitness Exercises by Reading Hand Palm

A. H. Elder Akpa , **Masashi Fujiwara**, **Hirohiko Suwa**, **Yutaka Arakawa** ,
and **Keiichi Yasumoto**

Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Nara, Japan

Correspondence should be addressed to A. H. Elder Akpa; akpaelder@gmail.com

Received 14 August 2018; Revised 24 December 2018; Accepted 13 February 2019; Published 21 May 2019

Academic Editor: Antonio Martinez-Olmos

Copyright © 2019 A. H. Elder Akpa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Medical studies have intensively demonstrated that sports activity can enhance both the mental and the physical health of practitioners. In recent years, fitness activity became the most common way to motivate and engage people in sports activity. Recently, there have been multiple attempts to elaborate on the “ideal” IoT-based solution to track and assess these fitness activities. Most fitness activities (except aerobic activities like running) involve one or multiple interactions between the athlete’s hand palms and body or between the hand palms and the workout materials. In this work, we present our idea to exploit these biomechanical interactions of the hand palms to track fitness activities via a smart glove. Our smart glove-based system integrates force-sensitive resistor (FSR) sensors into wearable fitness gloves to identify and count fitness activity, by analyzing the time series of the pressure distribution in the hand palms observed during fitness sessions. To assess the performance of our proposed system, we conducted an experimental study with 10 participants over 10 common fitness activities. For the user-dependent activity recognition case, the experimental results showed 88.90% of the F score for overall activity recognition. The result of leave-one-participant-out cross-validation showed an F score ranging from 58.30% to 100%, with an average of 82.00%. For the exercise repetition count, the system achieved an average counting error of 9.85%, with a standard deviation of 1.38.

1. Introduction

The last decades have seen a growing interest in systems that encourage and support people with regard to the regular practice of physical activity in general and fitness activity in particular. This keen interest in physical activity is mainly due to the results of many research studies which claim that regular practice of physical activity has positive effects on mental and physical health [1–3].

With the recent development of wearable computing and sensor technology, a wide range of ubiquitous systems has been developed by researchers and commercial industries to support and motivate people towards physical activity. These systems are designed to track physical activity, evaluate user results, and support decision-making to improve user performances. Fitness workout tracking is important because fitness tracking data provide practitioners with a sense of direction, help them adjust their workout routine, and enhance their motivation and willingness to change their lifestyle [4, 5]. However, the majority of these existing

ubiquitous systems either are limited to only track aerobic exercise (biking, running) [6, 7] or focus only on muscle and strength training (push-up, chest press) [8, 9]. In other words, each of these systems was designed to only track and evaluate a specific set of physical activities.

Thus, the use of such systems tends to limit or constrain the users to perform only within the predefined set of exercises.

For example, the system in [10] proposed the use of a smart mat to recognize and count gym exercises. Although the system achieved good results (recognition rate of 82.5% and counting accuracy of 89.9%) for exercises performed on the mat (such as push-ups, crunches, and squats), it appears obvious that the system cannot be used for gym exercises which are not performed on a mat (pull-up or bench dip). Our proposed system is motivated by the following two observations:

- (i) In general, fitness athletes (with no distinction of their gender, age, or goal) usually wear sports gloves during their workout sessions. They wear the sports gloves for various reasons such as supporting grip

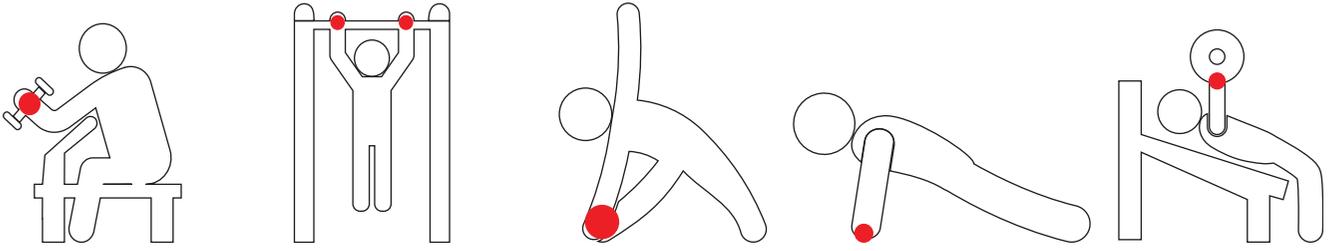


FIGURE 1: Examples of commonly performed fitness exercises.

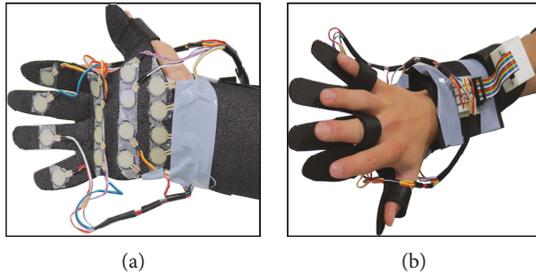


FIGURE 2: Inside and outside views of the smart glove prototype.

pressure, protecting the hands from calluses and blisters, or increasing lift power [11]

- (ii) During workout, for most fitness activities (except for aerobics activities like running), there is always one or multiple interactions between the gloves (hands) and the athlete's body or between the gloves and the workout materials

With that in mind, in this work, we propose a smart glove that can track fitness exercises whenever there is a direct interaction between the athlete's hands and the workout environment. Figure 1 shows some examples of commonly practiced fitness exercises with the red dots representing the areas of interaction between the hand palm and the environment.

Our proposed smart glove-based system (Figure 2) is designed with 16 force-sensitive resistor (FSR) sensors for tracking, recognizing, and counting physical activities performed in any fitness environment. It relies on a single sensorized wearable device to monitor a wider range of fitness activities than existing systems [12, 13], without the need to modify fitness environment or to attach various sensors on different parts of the athlete body.

The novelty of our proposed system resides in the approach and data type used to track fitness exercises. Indeed, most common fitness tracking systems rely on inertial sensors such as accelerometers to get information about users' body positions and movements, whereas our system utilizes data coming directly from the contact surfaces between the users' hands and the workout environment. Based on this approach, our system could be more suitable to distinguish between exercises of the same type. For example, a classic push-up (Figure 3(a)) will produce a pressure distribution different from the distribution of a cross-body push-up (Figure 3(b)) or a reverse hand push-up (Figure 3(c)). This kind of distinction among the same types of exercises become difficult when

using only inertial sensors because the body positions and movements are quite similar.

Our ultimate goal with the proposed system is to provide fitness athletes with a real-time system that

- (i) recognizes the activity being performed
- (ii) counts the number of repetitions (reps)
- (iii) estimates the calorie burned out by each exercise set
- (iv) recommends future exercises to achieve users' goal

In this paper, we present the design, the activity recognition, and the exercise counting performance of our smart glove system.

The core contributions of this paper are as follows:

- (i) *Novel wearable sensing approach*: we introduce a new resistive pressure glove that uses FSR sensors suitably mounted in the inside part of a sports glove to read real-time biomechanical information from the hand palm to recognize and count fitness activity
- (ii) *Single activity detection*: by analyzing the pressure distribution applied over the entire palm surface during a workout session, we can recognize most exercises of the following 4 different groups of fitness training
 - (a) flexibility training (e.g., side lunge stretch)
 - (b) dynamic strength training (e.g., squat)
 - (c) static strength training (e.g., plank)
 - (d) circuit training (e.g., push-up + bench dip + lunge)

Through an experiment study, we showed that our smart glove system successfully recognizes 10 frequently performed fitness exercises. We selected 10 different fitness exercise types that target different muscle groups as described in Section 4. Using the ensemble subspace k-NN classification method, over 300 exercise sets from 10 participants, the system achieved 88.00% recognition accuracy with the 10-fold cross-validation. In the case of the leave-one-participant-out cross-validation, we obtained an average recognition accuracy of 82.00%

- (iii) *Exercise counting approach*: for each exercise set, by investigating the signal from the glove's sensors,

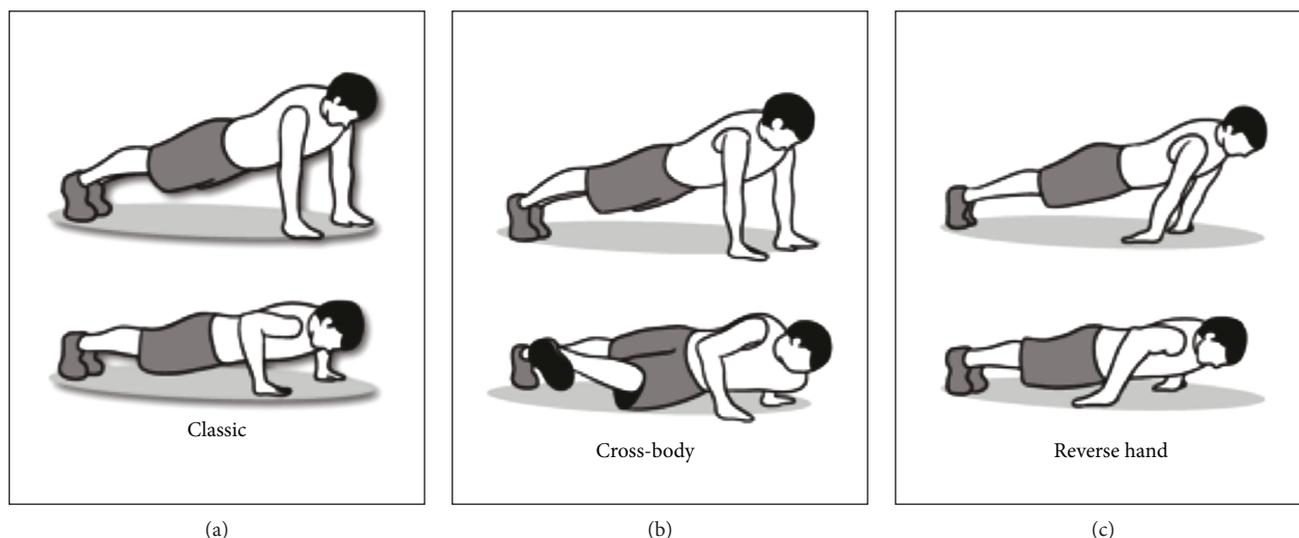


FIGURE 3: Three distinctive types of push-ups.

the system automatically counts the number of exercise repetitions without prior need of training templates. By doing so, we ensure that repetitions of each exercise set are counted independently of the activity type and the user's workout speed. We evaluated the exercise counting performance of the system using 5186 repetitions from the 10 experiment participants. The overall results show an average counting error rate of 9.85%

The remainder of this paper is organized into the following sections. Section 2 presents existing literature and related work. Section 3 describes the design and hardware of the smart glove. Section 4 gives the details about the experiment setup. Section 5 discusses the analysis and activity classification performance of the system, while Section 6 explains the counting algorithm with its results. Section 7 discusses some relevant points of this study, and finally, Section 8 concludes this paper and provides the future work.

2. Related Work

There have been much work and many systems on the use of technologies to support and track physical activities. In this section, we restrict the literature review to the area of commercially available systems and research work that use wearable sensors and sensorized equipment to monitor and assess fitness activities.

2.1. Off-the-Shelf Technology. Commercially available devices such as running watches, fitness wristbands, or trackers are nowadays ubiquitous tools used by thousands of people. Fitbit [16] and Garmin [17] are two of the most common wireless physical activity trackers present in the market. These systems use heart-rate sensors, GPS, pedometers, and other sensors to track physical activities. However, they mainly provide feedback and support for aerobic workouts (walking, running, or biking) and almost no feedback for strength and muscular training exercises. For example, Fitbit does not

recognize non-step-based activities (such as push-ups), and it requires the user to manually log strength training such as dumbbell curl [18].

Another well-known commercial system is ActiveLinxx (formerly known as Fitlinxx) [19]. This system integrates multiple sensors on strength training machines to track the load of weight a user is moving, count the number of repetitions, and show training progress on the built-in display of each machine. Although the new version supports syncing with users' smartphone or tablet, this system still requires the fitness center owners to purchase and install these training machines.

Google Fit is another widely used activity tracking system which is in the form of an application. It is available for download on any Android device and uses the built-in sensors of that device to track commonly measured fitness data such as the number of steps and calories burned. However, Google Fit's support website revealed some limitations of the system [20]. For example, it is reported that walking, running, and biking detection is not perfect on Google Fit, because smartphone or tablet sensors are not accurate enough for tracking this type of activities.

2.2. Researches in Fitness Tracking

2.2.1. Glove-Based Systems. As for research works that use gloves for tracking physical activities, Chang et al. [12] presented a system to track free-weight exercises, by incorporating a 3-axis accelerometer into a workout glove and another accelerometer on users' waist. They used a naive Bayes classifier and hidden Markov models to recognize the type of exercise. To count repetitions, they developed a peak counting algorithm and a method using the Viterbi algorithm. Their experimental results showed good results with a recognition accuracy of 90% and error count of around 5% over 9 different exercises. The limitation of this work is that they focused only on free-weight exercises such as the bench press, biceps curl, and lateral raise.

Another work presented by Ye et al. [14] proposed a force-sensing glove system for measuring real-time hand forces during motorbike riding activity with the aim of giving feedback to the riders. Their prototype device consists of two gloves with 4 tactile sensors (A401 FlexiForce) attached inside each glove and connected to a microcontroller through front-end electronics. Contrary to our work, this system focuses on different actions (clutch, throttle, brake, and steer) observed during a single outdoor physical activity.

2.2.2. Pressure Sensing in Fitness Activity. Sundholm et al. [10] proposed a textile pressure sensor matrix that can be integrated into fitness mats to recognize and count strength related exercises. Their experiment with 7 participants showed that the pressure sensor mat successfully distinguished 10 strength exercises with a recognition accuracy of 82.5% using a k-NN classifier and counting accuracy of 89.9%. The major limitation of this system is that it cannot be used for gym exercises which are not performed on a mat. Our work complements the areas not covered by their system with a glove that reads and analyzes hand palm interactions even for exercises that are not performed on a mat.

A different work introduced by Zhou et al. in [15] described a wearable textile sensor system for monitoring only gym leg exercises. The system relies on surface pressure changes between the quadriceps skin and an elastic sports band to track and evaluate gym leg workout. In an experiment with 6 participants and 24 leg workout sessions, they achieved 81.7% of average recognition accuracy. More works involving the use of pressure distribution were proposed to contribute to the research field of sports activity assessment, in general. Zhou et al. [21] made a smart soccer shoe that uses pressure sensing matrices to detect and analyze the interaction between the player's foot and the ball. Instead of using add-on sensors to the shoe, they integrated the pressure sensing element inside the shoe surface material in an unobtrusive fashion that can be manufactured together with the shoes. The sensor system consists of two 34 and one 33 pressure sensing matrices. Their experimental best performance nearly reaches 100% accuracy for 15 different types of ball shots.

2.2.3. Other Sensor-Based Fitness Systems. A state-of-art study, similar to our work is the "RecoFit" system developed by Microsoft Research [8]. RecoFit uses inertial sensors attached to the upper forearm of the athlete and measures a 3-axis accelerometer and gyroscope to recognize and count weight training and calisthenics exercises. The RecoFit system gives classification accuracy of 100%, 99.3%, 98.1%, and 96.0% for 4, 4, 7, and 13 types of exercises, respectively. The exercise types included exercises such as squat, crunch, push-up, shoulder press, triceps extension, and back fly. Although this result is impressive, there are several points to remark for comparison to our work. First, RecoFit uses only an autocorrelation method to eliminate peaks in the signal that do not correspond to actual exercise repetitions, whereas we utilize 3 methods including dynamic time warping (DTW) to get the actual repetitions. Furthermore, our work focuses on exercises of 4 major groups of fitness

training, while the RecoFit system focuses on 2 exercise groups which are weight training and calisthenics.

Another more recent work developed by Hassan et al. [6] called "FootStriker," assists runners by actuating the calf muscles during treadmill running sessions. FootStriker is a wearable system that detects the user's running style using force-sensitive resistors (FSR) in the insole of a running shoe and uses electric muscle stimulation (EMS) as a real-time feedback channel to intuitively assist the runner in adopting a mid- or forefoot strike pattern. Their experiment results showed that EMS actuation significantly outperforms traditional coaching systems. While this study is closely related to ours in terms of using FSR sensors, it was mainly designed to detect a running strike and provide real-time feedback without recognizing and/or counting the strike to assess the runner performance.

To cope with the limitations of the aforementioned systems, we propose a new type of a smart glove-based system to assess fitness exercises by analyzing hand palm information during workout sessions. Our approach does not require readjusting existing layouts of fitness centers or buying new fitness machines. Table 1 gives a summary of the related studies, along with the comparison with our proposed smart glove system in terms of target exercises, sensors used, and type of data collected.

3. System Design and Architecture

This section presents the design and architecture of the smart glove system as described in our previous work [22]. The system intends to record and analyze the pressure distribution applied to a user hand palm during his/her training sessions. Therefore, we designed our prototype by taking into consideration the following technical aspects:

- (i) *Sensing zone:* a smart glove that covers well the palm surface will enable us to get the accurate location of high pressure points and the pressure distribution applied over all the palm surface
- (ii) *Portability:* the device should work in any fitness center or workout environment. Also, it should be lightweight and easy to wear
- (iii) *Comfort level:* comfort is an important aspect when designing an IoT device worn by human users. Therefore, the glove should provide an acceptable or high comfort level to users, without hindering users' movement

The current prototype of the smart glove system consists of 3 main components: a set of 16 FSR sensors, a data sampling unit (DSU), and a visualization and computation software.

3.1. The Force-Sensitive Fitness Glove. The smart glove contains a set of 16 force-sensitive resistor (FSR) sensors which are mounted on the palm of the glove as depicted in Figure 4. The 16 FSR sensors are used to read the forces applied to the palm when a user is performing a fitness

TABLE 1: Comparison with some related work.

Related system denomination	Targeted exercises	Type of sensors used	Collected data type	Classification/counting
Tracking free-weight exercises [12]	Weight training exercises	3-axis accelerometers	Acceleration	Yes/yes
Force-sensing glove system for measurement of hand forces during motorbike riding [14]	Motorbike riding	FlexiForce tactile sensors	Tactile forces	Yes/no
Smart mat [10]	Strength and muscular endurance	Resistive pressure sensor matrix	(a) Body weight distribution (b) Pressure distribution	Yes/yes
Never skip leg day [15]	Leg exercises	Textile FSR sensors	Pressure distribution	Yes/yes
RecoFit [8]	(a) Weight training (b) Calisthenics	3-axis accelerometer and 3-axis gyroscope	(a) Acceleration data (b) Rotational motion data	Yes/yes
Our proposed system	(a) Flexibility training (b) Dynamic strength training (c) Static strength training (d) Circuit training	FSR sensors	Pressure distribution	Yes/yes

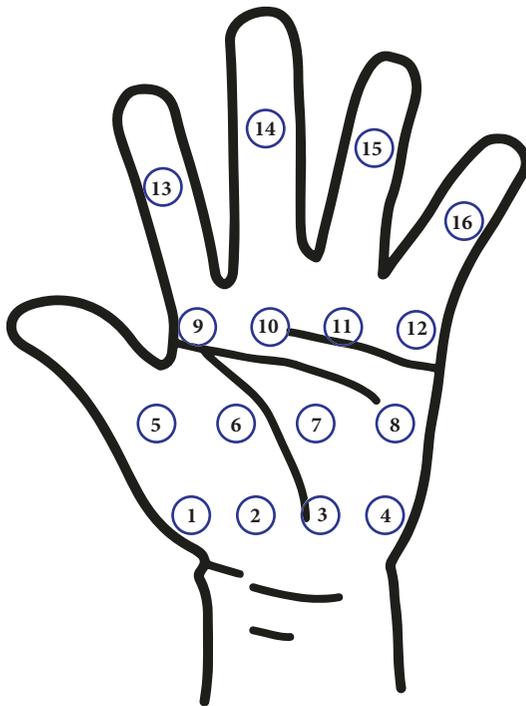


FIGURE 4: The sixteen locations on the glove where the force-sensitive sensors are positioned to read the pressure distributions.

exercise. FSR sensors are commonly used to detect physical pressure, squeezing, and weight. They are easy to use and low-cost. In general, FlexiForce and Interlink are two common types of FSR sensors that are available, cheap and easily found in the market. Studies have shown that the FSR is usually applied for robotic grippers and for biomechanical fields [23]. In this work, because of our design criteria we chose the Interlink 402 Short Tail model [24] over the most popular FlexiForce A201 [25]. Indeed, to avoid making a bulky or uncomfortable prototype to wear, we favored the use of the Interlink 402 of length 6.03 cm over the FlexiForce A201, which has a total length of 10.16 cm.

Each sensor has an active sensing area of 12.70 mm diameter, with 0.46 mm thickness, and can sense forces up to 20 N. Based on the force applied to the sensors, their electrical resistive values change and the voltage values corresponding to the pressure is read by the DSU. All sensors are separately connected to the DSU using twisted pair cables (Figure 2).

For now, we use only one hand of the fitness glove pair. For this first prototype of our fitness glove, we wanted to ensure the maximum or full coverage of the athlete's hand palm, in order to get as much as possible pressure distribution data from different areas of the palm. For this reason, we utilized 16 FSR sensors on our first prototype. This number of sensors might be excessive, but it guarantees full coverage of the sensing zone, which is important when studying a first prototype of a wearable sensing device. However, for our final system, we intend to reduce this number of FSR sensors based on the results and observations of this work.

3.2. The Data Sampling and Communication Unit (DSU). The data sampling and communication unit (DSU) reads the data from the force sensors and sends the data to the computing and visualization unit. The DSU is composed of an Adafruit Feather 32u4 Bluefruit, a 16 channel multiplexer (16chMUX), a resistor of 2 k Ω , and a small 400 mA/h Li-Po battery.

The Adafruit Feather 32u4 Bluefruit is Arduino-compatible+Bluetooth Low Energy (BLE) with a built-in USB and battery charging module [26]. It has a 10-bit resolution of an AD converter and can read raw digital values from 0 to 1023. The DSU samples and transfers sensing data via BLE at 5 Hz. The operating supply voltage is 3.3 V, and it is powered by a small 400 mA/h Li-Po battery. Our battery-life duration tests show that it can last over 6 hours, which is enough for weekly use, considering that average gym-goers train 3 days per week for about 2 hours per training session [27]. The DSU components are connected and put inside a white 3D-printed box and positioned on the athlete's forearm. The DSU and the 16 FSR sensors constitute the hardware portion of the smart glove system that the athlete wears during his/her training session. This hardware portion

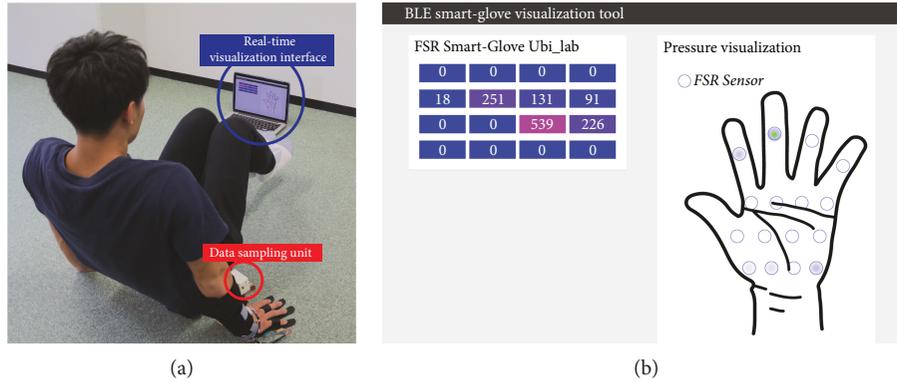


FIGURE 5: (a) A user performing a knee-pull-in exercise. The real-time visualization laptop and the DSU are indicated by the blue and red circles, respectively. (b) Screenshot of the visualization UI tool for an exercise.

weighs approximately 101 g (16 sensors + wires \approx 43 g and the DSU with 3D case + battery \approx 58 g).

3.3. Visualization and Computation Software. The visualization and computation software is responsible for logging and displaying the data sent via Bluetooth by the DSU. It provides, through a web-based user interface (UI), a real-time feedback of the activity's pressure distribution, with the option to save the data into a CSV file (Comma-Separated Values) for further processing. The value of the pressure intensity applied to each sensor is indicated as a heat map-like color distribution on the glove image on the display. Figure 5(a) shows an example of a user doing a knee-pull-in exercise while watching the visualization UI tool (Figure 5(b)).

4. Experiment

To analyze the system's capability of recording, classifying, and counting fitness activities, we designed and ran a 1-hour workout session experiment with 10 participants. Participants were informed that they could stop the experiment at any time without losing benefits. Before their enrollment and participation, we obtained written informed consent from each participant. With participants' consent, we recorded video of the experiment that we used later as ground truth data to evaluate the performance of the system. Out of 40 preselected fitness exercises, we selected 10 exercises for our experiment based on the following two criteria:

- (i) *Exercises with less risk of injuries:* since our participants are graduate students with their own academic schedule, we prioritized exercises with minimum injury risk
- (ii) *Exercise group that targets full-body muscle training:* most fitness programs provide workout plans that focus on a full-body basic set of muscles which are the biceps, triceps, chest, legs, and abs [28]. Therefore, we chose to focus on common exercises suitable for both amateurs and professionals that work out these full-body muscles

4.1. Participant Demography. For the experiment, we recruited 10 healthy participants aged from 22 to 30 years (mean = 25.9, standard deviation (SD) = 3.21). Their weight ranged from 50 kg to 82.2 kg (mean = 67.2, SD = 10.07) and height between 158 cm and 182 cm (mean = 173.4, SD = 6.32). Only 1 out of the 10 participants was a female. All participants were full-time graduate students with different workout levels and experience. Five participants were regular gym-goers, who usually exercise 3.5 times/week. Three participants do not often exercise and did not practice any sport in the last six months before the experiment day. The two other participants reported exercise once or twice a week when they have time but not on a regular basis. Three participants were also participants of our previous study about users' requirements for a smartphone fitness application, and they were highly motivated to see how our smart glove could help them to improve their workout performance [29].

4.2. Calibration. Before their application, it is necessary for force-sensing systems to be calibrated for reducing inaccuracies. Since the resistivity values of the FSR sensors mainly depend on the person's weight, the curved shape of the hand palm, and the sensors' position, calibration is needed for each participant. Like with many related works using garments [6, 14], in this system, we also calibrated our smart glove sensors under static conditions. Prior to the start of the experiment, for calibration purposes, we asked each participant to do a 30-second push-up as his/her warm-up session. The data collected during this warm-up period were used to define for each participant the higher and lower threshold values of the sensors. The lower threshold corresponds to the highest value below which the sensor is considered "not activated," which means there is truly no pressure on this point. The higher threshold is the lowest values above with the sensor is considered "fully activated," which means there is pressure on this point.

Between the end of the warm-up session and the beginning of the experiment, we observed at least a 2-minute break to eliminate any influence of sensor drift and allow the participants to be ready.

4.3. Experiment Setup and Tasks. The participants were asked to perform the 10 selected fitness exercises while wearing the

TABLE 2: Selected fitness exercises with their target muscle group.

	Exercises	Target muscle group
1	Bench dips	Triceps, shoulders
2	Climber	Hips, legs, quadriceps
3	Dumbbell curl	Biceps
4	Knee-pull-in	Abdominals
5	Knee-twist-in	Abdominals
6	Plank leg raise	Lower back, glutes, triceps
7	Pilate dips (triceps)	Triceps, biceps, shoulders, back
8	Push-up	Chest
9	Side-to-side lunge	Glutes, quadriceps, butt
10	Wall push-up	Arms, shoulder, chest

smart glove. Each exercise is executed 3 times (3 sets for each exercise), at the participant's natural speed and pace, to make the experiment more realistic. Each set lasts for 30 seconds, and the participants were allowed to take a 1- or 2-minute break between the sets. During the break time, the glove was turned off by the participants to save battery. The ordering of the exercise was randomized, each participant freely chose the order in which he/she wanted to perform the exercise. Table 2 lists the 10 exercises along with their corresponding target muscle group, and Figure 6 shows a sketch of each exercise with the red area representing the interaction zone between the glove and the environment. Depending on each participant's pace, the overall experiment lasts for 1 hour to 1 hour and a half.

4.4. Procedure and Apparatus. At first, each participant is asked to read and sign the informed consent statement. Then, we introduced the experiment as well as the 10 workout exercises to the participant. After filling out the demographic data sheet, the participant is equipped with the smart glove on his/her dominant hand (left or right hand), with nothing attached on the other hand. After that, we ask the participant to warm-up for 3 minutes by freely doing any stretching movements he/she wants. During the experiment, all participants wore the smart glove and the DSU unit to read the pressure distribution on the hand palm and send the data via Bluetooth to the visualization software for displaying and saving of the workout data. If needed, participants were allowed to remove the smart glove during the break time between sets. Once the experiment is over, we ask each participant to take a short after-experiment survey of 2 questions:

- (1) How comfortable is the smart glove (rate on a 10-star scale)
- (2) Any feedback, comments, suggestions, or ideas related to the device and the system

In addition to the smart glove apparatus, we provided participants with a 2 kg dumbbell to perform the dumbbell curl exercise.

5. Fitness Activity Recognition: Methods and Results

The raw data from the smart glove is a stream of time series data of the 16-channel pressure values (each sensor represents a channel). From these pressure values, we extracted a set of features that could be used to distinguish between fitness exercise types. To analyze the pressure values and develop the classification model, we wrote several MATLAB scripts to select the features, compute the features, and validate the obtained results. MATLAB functions such as mean frequency (meanfreq) or root-mean-square (rms) have been exploited. All statistical analyses were performed using the MATLAB software.

5.1. Feature Selection and Extraction. For each exercise set performed by a user, we characterize the exercise as a single signal, denoted by $weight(t)$, obtained by averaging the 16-channel values (different values from each channel) over the time. In other words, we represent each exercise set with a single signal, by "grouping" the 16 channel signals.

The signal $weight(t)$ is the mean of the 16 channels computed over each row of the time series data. It is represented as a column vector and is calculated using

$$weight(t) = \frac{\sum_{i=1}^{16} channel_i}{16}. \quad (1)$$

The computed $weight(t)$ is then normalized according to its mean value and standard deviation. Hereinafter, we will use the notation $W(t)$ to refer to the normalized $weight(t)$.

All the activities provoke different changes in the pressure distribution and the intensity, at different points on the palm surface, in the temporal domain. Therefore, we computed the following set of time and frequency domain features:

- (i) Mean of each channel
- (ii) Standard deviation for each channel
- (iii) Number of above mean crossing of $W(t)$
- (iv) Number of below mean crossing of $W(t)$
- (v) Number of peaks of $W(t)$
- (vi) Skewness of $W(t)$
- (vii) Kurtosis of $W(t)$
- (viii) Band power of $W(t)$
- (ix) Mean frequency of $W(t)$
- (x) Max power spectrum of $W(t)$ FFT

Overall, 40 features were extracted (32 directly from the sensors and 8 from the signal $W(t)$) and used to train the classification model. Many of these features have been intensively investigated in previous studies and proved to be effective for activity recognition [30, 31]. For example, the mean crossing feature has been heavily used in human speech

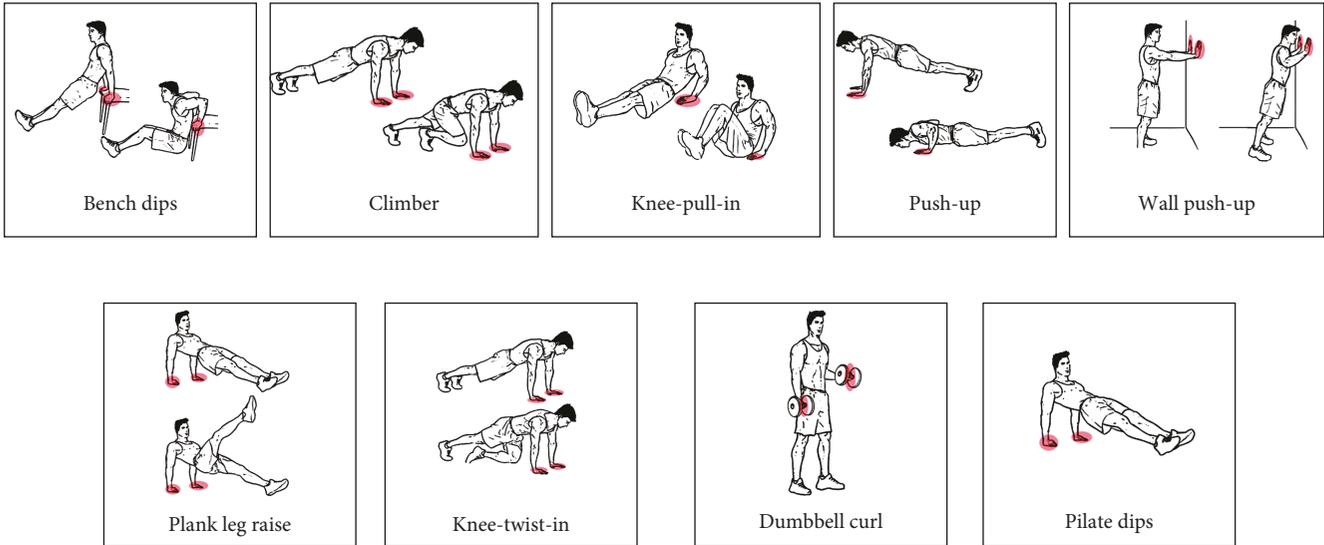


FIGURE 6: Exercises performed during the experiments, except the side-to-side lunge. Red areas represent the interaction surface between the glove and the workout environment.

recognition and handwriting recognition problems [30]. Statistical features such as the mean or kurtosis are frequently used because of their simplicity and high performance across several human activity recognition problems [32].

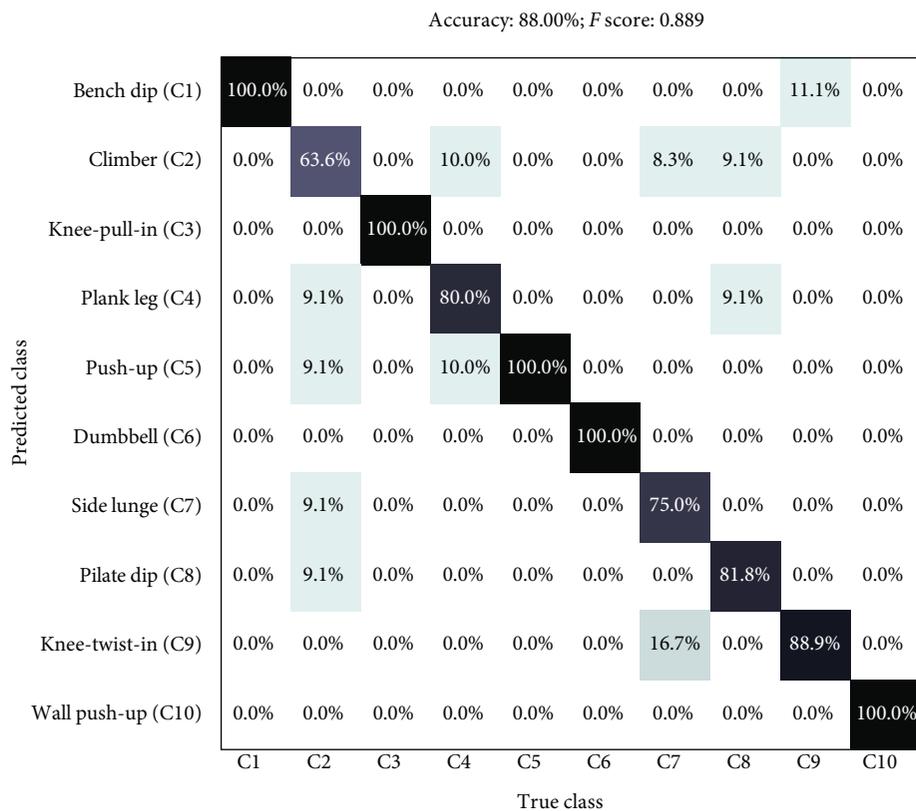
5.2. Recognition Results. To develop our classifier, we tested various classification algorithms, particularly the decision tree, random forest, SVM, k-NN, and ensemble methods. We report the result for the classifier based on the ensemble subspace k-NN method, which achieved the best recognition results. It is well-known that ensemble methods can be used to improve better predictive performance than could be obtained from any of the constituent learning algorithms alone [33]. To assess the robustness of the classifier against known and unknown users, we performed the following two evaluations:

- (i) *“Leave-one-session-out” cross-validation:* as a baseline, we first evaluated the recognition performance for every single participant. For each participant and each activity dataset, we trained the classifier with 2 out of 3 exercise sets and tested the classifier on the remaining exercise set. The person dependent evaluation was repeated for all 10 participants, with 10-fold cross-validation on each dataset. The confusion matrix of the average recognition results along with the F score is presented in Figure 7(a). The overall exercise recognition accuracy nearly reached 88.00% more precisely, with 6 of the 10 exercises having a recognition rate of 100%. The lowest misclassification of 53.80% was observed with the climber exercise. The F score was 0.889. It is worth pointing out that only three exercises (pilate dip, plank leg raise, and climber) have their recognition rate lower than 80%

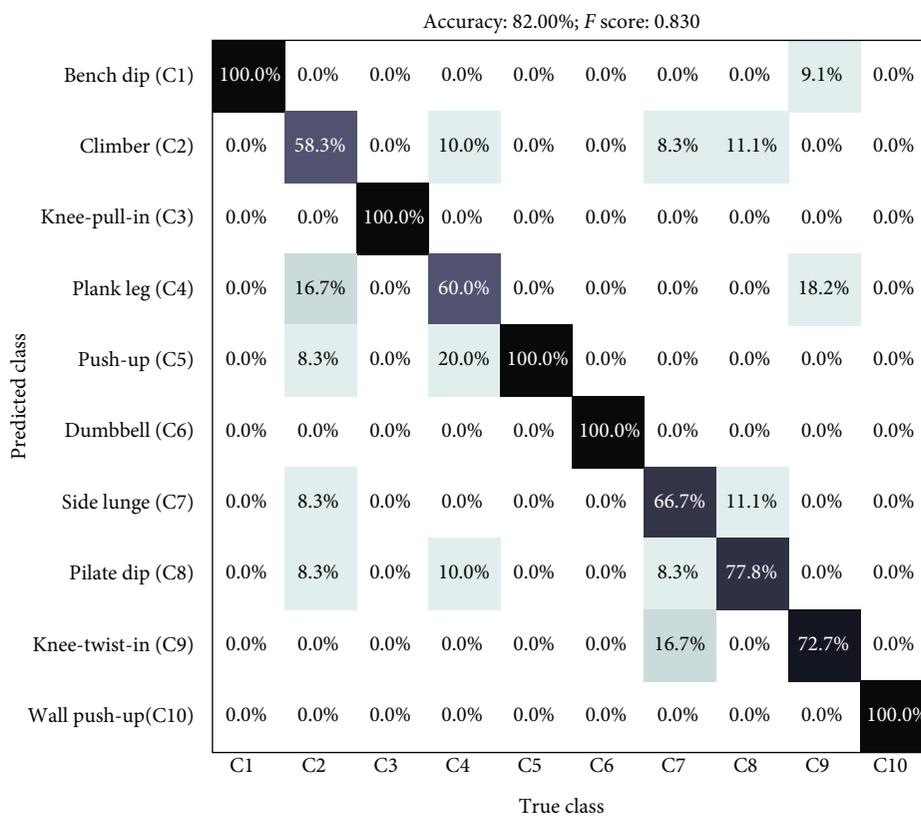
- (ii) *“Leave-one-participant-out” cross-validation:* ideally, the system should be operational for any new user without additional training procedure. We, therefore, performed leave-one-participant-out cross-validation to understand the robustness of the system against new unregistered users. For each participant, the classifier is trained with all the data from the other participant and it is tested with the data from that participant. The confusion matrix of the average recognition results and the F score are shown in Figure 7(b). The person-independent result shows an overall accuracy of 82.0%, with an F score of 0.830. This suggests that our system works well across different users. In this case, also, the climber and plank leg raise exercises have the lowest accuracy rates of 58.3% and 60%, respectively. Five exercises have their recognition rate lower than 80% (pilate dip, knee-twist-in, side lunge, plank leg raise, and climber)

For both evaluations (leave-one-session-out and leave-one-participant-out), the overall recognition results indicate highly acceptable levels of exercise recognition accuracy which are higher than 80%. Most of the misclassification happens with exercises such as climber and plank leg raise. We also note an inaccurate prediction for pilate dips, where athletes keep the same posture till the end of the exercise.

The decrease in the accuracy rate between the person-dependent and the person-independent evaluation is undoubtedly a result of participants having different workout styles, hand sizes, etc. We believe the system could be more robust against new users if we provided more training data that include more participants. As smart gloves are intended for personal usage, we suppose the optimum recognition accuracy can be obtained for end users that agree to provide training data during their first use.



(a) Average recognition result for leave-one-session-out



(b) Average recognition result for leave-one-participant-out

FIGURE 7: Average recognition result for (a) person-dependent and (b) person-independent evaluation.

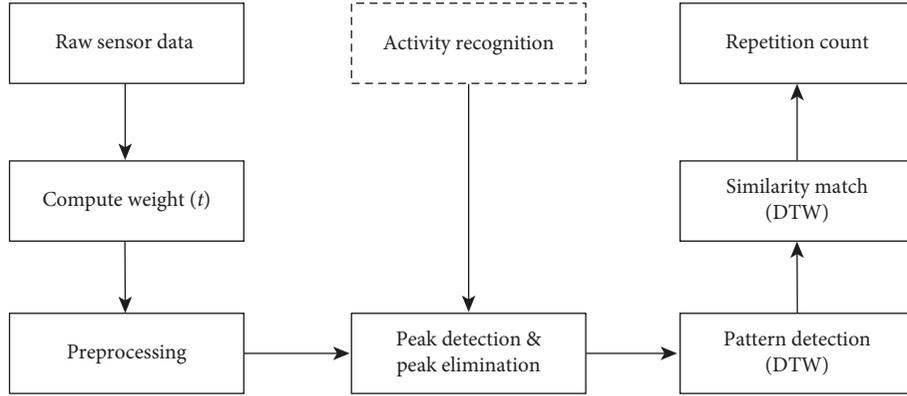


FIGURE 8: Overview of the DTW-based repetition count algorithm.

TABLE 3: Algorithm notations.

Notation	Description
$W(t)$	The mean of the 16 sensors for each time sample
Δ_{period}	The minimum possible time to perform a repetition
SZ	Searching zone of 10 seconds to find the repetition pattern
CP	Candidate peaks obtained after Step 2
RC_j	Repetition candidate at index j
R_{pat}	Repetition pattern of an exercise set
$\text{distanceDTW}_{\text{pat}}$	Smallest normalized distance between matching RC
thresh_dtw	Threshold value used to accept or deny matching

6. Counting Repetitions: Methods and Results

One goal of our proposed smart glove system is to count how many repetitions a user has performed during an exercise set. To fulfill this goal, we designed a counting algorithm that uses the raw data from the smart glove to count the number of exercise repetitions during fitness sessions. An exercise repetition is characterized by a specific pattern, observed within the time series signal $W(t)$ obtained from the 16 force-sensitive resistor (FSR) sensors. The counting phase itself is composed of 3 steps:

- (i) *Step 1*: conditioning of the signal $W(t)$
- (ii) *Step 2*: peak detection and elimination
- (iii) *Step 3*: repetition pattern detection and similarity match

Figure 8 depicts the different steps of the counting algorithm. The dashed box indicates the activity type (exercise name) obtained from the activity recognition step described in the previous Section 5. The counting algorithm notations along with their descriptions are summarized in Table 3.

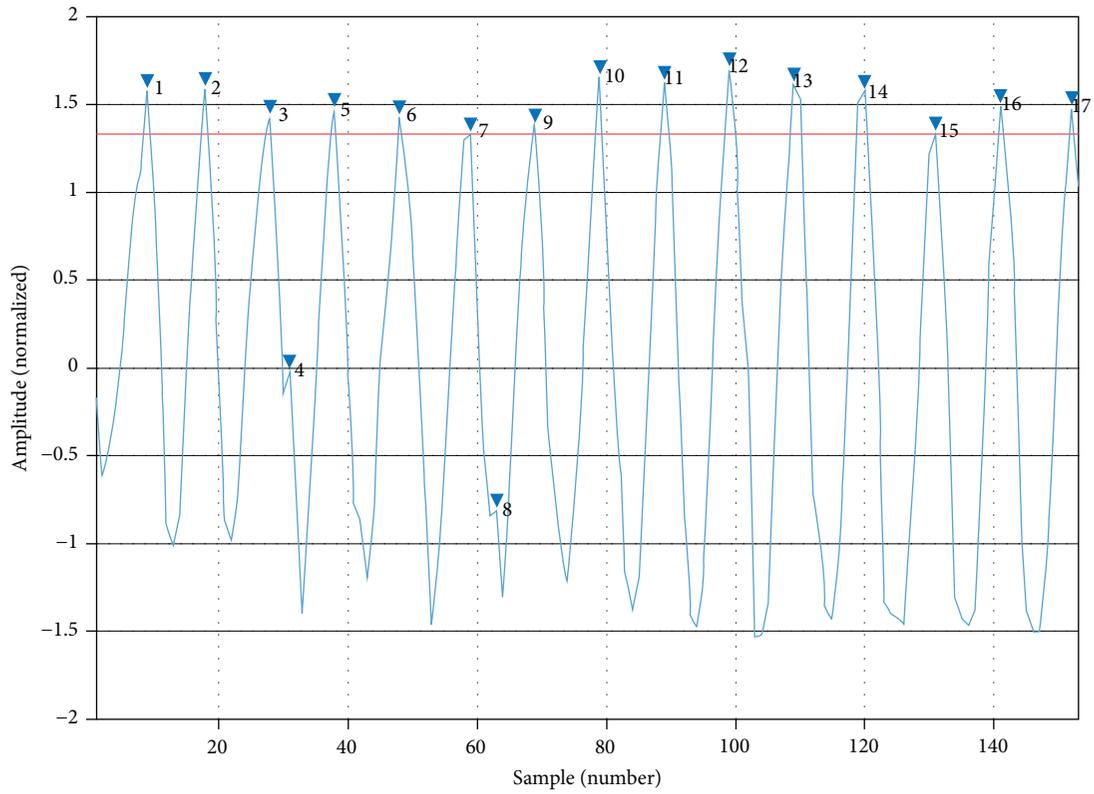
6.1. Weight Signal Conditioning. The signal $W(t)$, as explained in the previous Section 5.1, is the mean of the 16 channels computed over each row of the time series data. $W(t)$ is computed using equation (1) and normalized according to its mean value and standard deviation. For each exercise set, the acquired $W(t)$ signal is smoothed with an eleven-point fourth-order Savitzky-Golay (SG) smoothing filter [34]. The SG filter is a digital filter that can be applied to a time series data to smooth the data, that is, to increase the signal-to-noise ratio without significantly distorting the signal. The role of the SG filter is to approximate the values within a specified window by a polynomial of a specified order to minimize its least-square error. The advantage of using SG filtering before processing a signal is that it does not delay the signal and can preserve features such as local minima and maxima (peaks) [13]. Finally, the smoothed signal is passed to the next step (Step 2), for peak detection and elimination.

6.2. Peak Detection and Elimination. In this step, we exploit the local maxima (peaks) of the signal $W(t)$ to find repetition candidates and to reject peaks that are significantly low to be candidate peaks. The intuition behind our approach is that most of the exercise repetitions result in strong peaks.

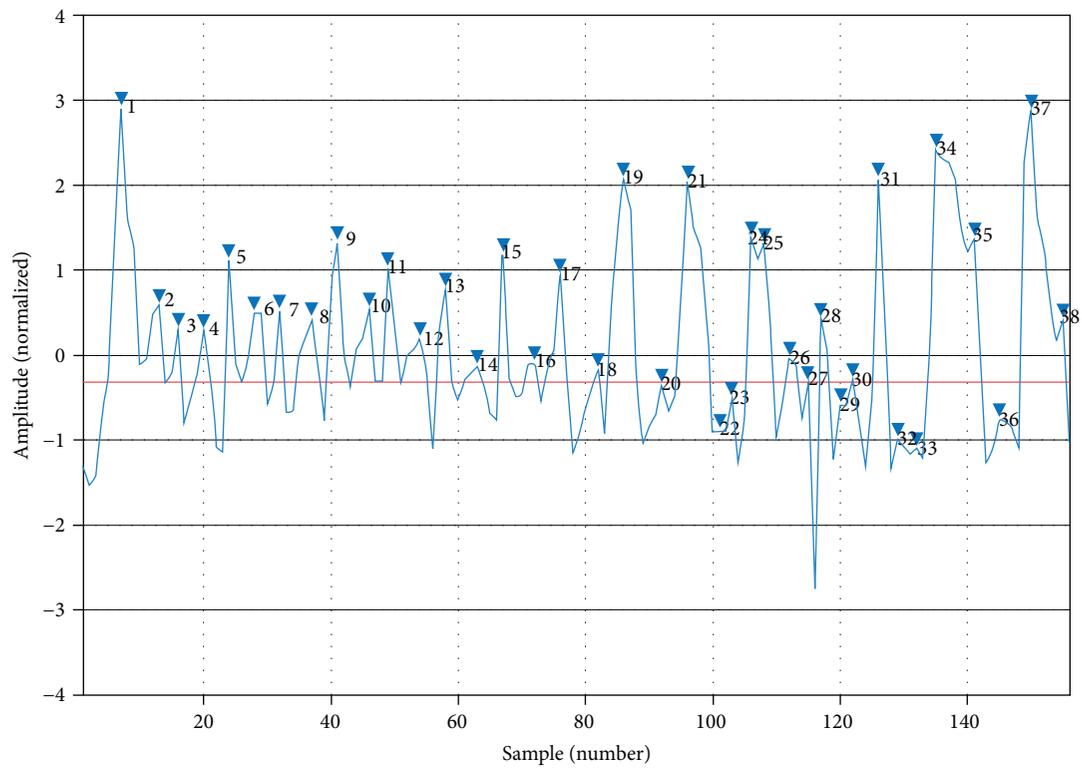
Figure 9(a) shows an example of a “simple-and-easy” signal, where each repetition corresponds to a high-amplitude peak, occurring at a regular time interval, with almost similar shapes. Counting the number of repetitions from such a fitness exercise signal is quite easy (the numbers and the blue triangles represent all the candidate peaks before running the counting algorithm). However, the signals obtained from the smart glove are not always that simple. Depending on the exercise type and the user’s workout style, the peak amplitudes and shapes are different and inconstant (Figure 9(b)).

Therefore, our algorithm has to be efficient and global to handle any exercise repetition signal from the smart glove. To that end, in this step, we need to find and reject peaks that are not generated by actual repetitions such as peaks caused by fatigue and subrepetitions.

To filter out false repetition peaks, we primarily compute all the local maxima of the $W(t)$ signal to obtain a list of all candidate peaks. Then, we sort these candidate peaks based on their amplitude and compute the peak at the 20th percentile. Secondly, we reject all peaks with an amplitude smaller



(a) Example signal with constant peak amplitude and regular shapes



(b) Example signal with inconstant peak amplitude and irregular shapes

FIGURE 9: Examples of repetition signals.

than the 20th percentile. The 20th percentile represents the amplitude value below which 20% of the candidate peaks may be found. That means 80% of the candidate peaks are higher than this value. The percentile-based technique has been largely used for filtering peaks in signal processing problems [35]. It appears that the percentile method is more efficient than setting absolute threshold values, as demonstrated in [8]. The reason is that setting a fixed lower threshold value will lead to a selection of many candidate peaks but might also contain multiple false candidate peaks; while a fixed higher threshold will only detect high-amplitude peak repetitions. Therefore, applying an appropriate n -th percentile technique appeared to be a good technique for 1st-level peak filtering. In our study, generally, peaks corresponding to fatigue or subrepetitions fall below the 20th percentile peak. In Figures 9(a) and 9(b), the red line represents the 20th percentile of each signal.

Finally, after rejecting the candidate peaks lower than the 20th percentile, we filter the remaining candidate peaks based on the time elapsed between successive candidates. We compare the time between a peak candidate and the previously accepted candidate. If the current peak candidate is at least Δ period seconds away from the previously accepted candidate, we accept this current peak candidate. If not, we reject the current peak candidate and move to the next candidate. The intuition behind this approach is that if two consecutive peaks in the signal are very close (e.g., less than 0.4 sec), one of them is not an actual repetition peak.

The value of Δ period (in seconds) represents the minimum possible time for a human to perform a repetition. For each exercise, we get this period based on the fastest repetition observed during our experiment for that exercise. For example, for the lateral stretch exercise, Δ period is equaled to 1.0 sec, and for push-up exercises, it is equaled to 2.0 sec.

After rejecting the false candidate peaks based on the 20th percentile threshold and the time elapsed, the remaining candidate peaks are passed to the next step (Step 3) for the repetition pattern detection.

6.3. Repetition Pattern Detection and Similarity Match. At the end of the two previously described steps, we already have a naive repetition count algorithm. We could have stopped the algorithm after these two steps. However, this algorithm would be sensitive to variations in timing, speed, and individual repetition style. Therefore, to make our counting algorithm robust against exercise type and speed variations, we incorporated the dynamic time warping (DTW) algorithm to find the repetition pattern of the exercise set, and then, we examined the signal $W(t)$ to find out how many times the same pattern is repeated in the signal.

Dynamic time warping (DTW) is a widely used technique that utilizes dynamic programming to find the optimum distance between time series [5]. It allows us to compare signal sequences that are not temporally aligned by creating a mapping which minimizes the distance between input sequences. The DTW algorithm has been proven to be effective for speech recognition [36] and has been intensively used in previous studies related to activity recognition [32], human motion [37], and bioinformatics [38].

In our counting algorithm, we utilize DTW to

- (i) find the repetition pattern (RP) of an exercise set (phase 1)
- (ii) measure the similarity between a repetition candidate (RC) and the repetition pattern (phase 2)

Compared to some previous studies as in [10], rather than selecting a standard template for an exercise type, we extract the particular repetition pattern of each single exercise set. By doing so, we ensure that each exercise set is evaluated independently of the activity type and the user's speed.

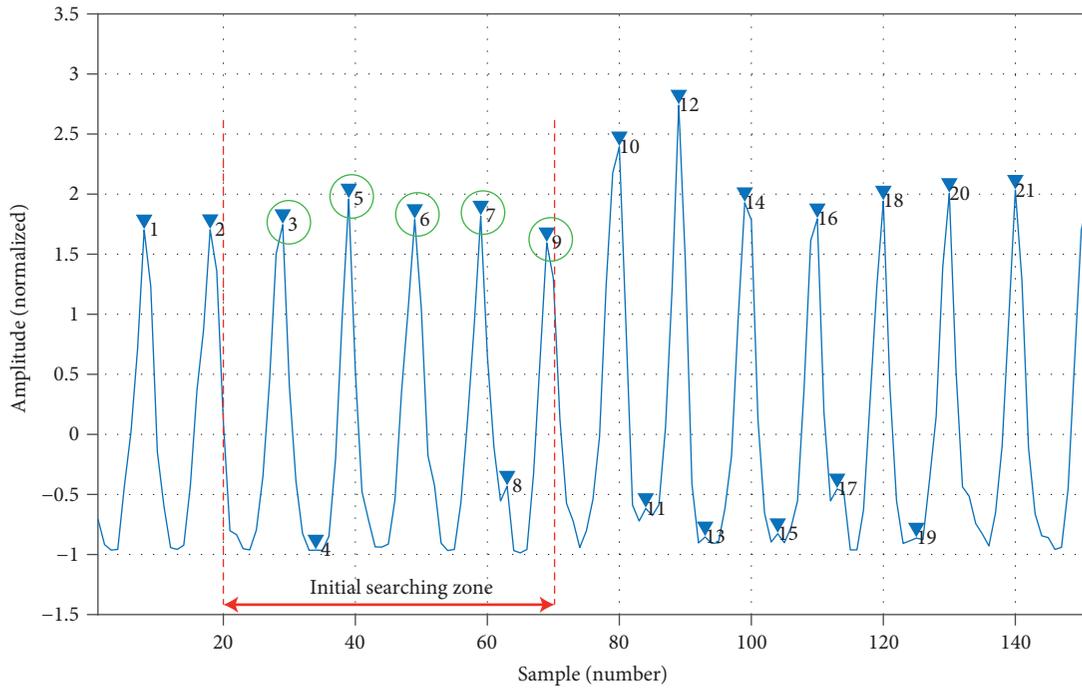
6.3.1. Find the Repetition Pattern (Phase 1). In this phase, we empirically defined a time window of length 10 seconds starting from second 4 to second 14 of $W(t)$. We called this time window the searching zone (SZ). The algorithm will look for the repetition pattern only inside the SZ. The reason behind starting from the 4th second is that most people usually utilize the first seconds of an exercise to adjust their body and/or find their comfortable hand position. For this reason, the first repetitions of an exercise are not always actual exercise repetitions.

We work with signal sequences between the consecutive candidate peaks (CP) obtained after Step 2. We denoted by RC_j (repetition candidate j) the signal sequence of $W(t)$ between the candidate peaks CP_i and CP_{i+1} , inside the SZ (Table 3). The algorithm loops through the repetition candidate and uses the DTW algorithm to find the first repetition candidate that matches the other repetition candidates inside the SZ. For example, given an exercise signal with 4 candidate peaks at indices CP_3 , CP_4 , CP_5 , and CP_6 inside the searching zone, the algorithm will generate 3 repetition candidates such as $RC_1 = W(t)[CP_3 : CP_4]$, $RC_2 = W(t)[CP_4 : CP_5]$, and $RC_3 = W(t)[CP_5 : CP_6]$, respectively. Then, we use the DTW algorithm to compare RC_1 to RC_2 , RC_2 to RC_3 , and RC_1 to RC_3 .

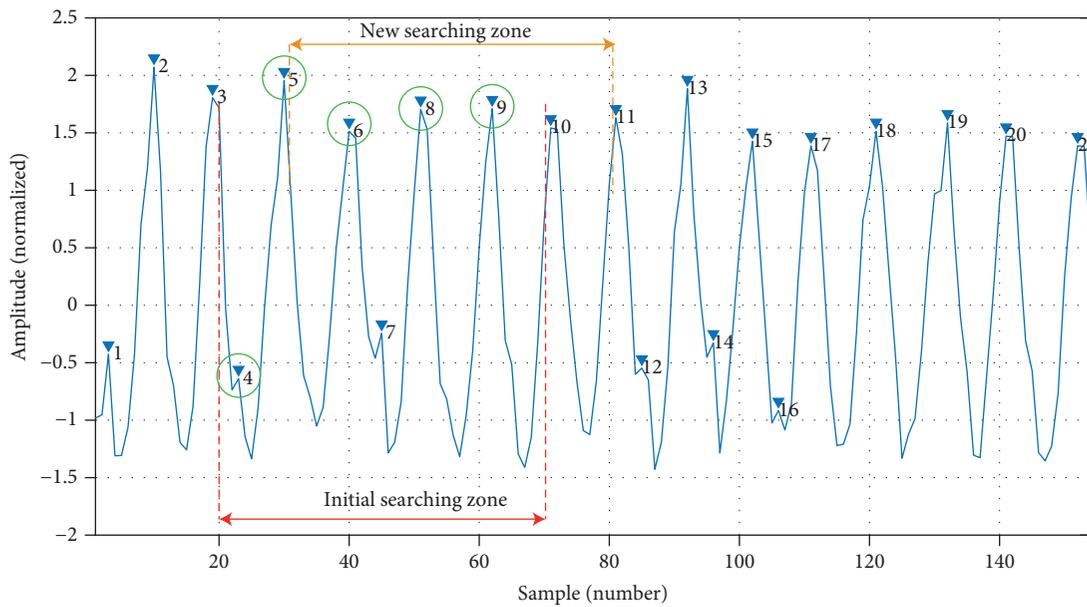
If RC_1 matches RC_2 and RC_3 , then the algorithm will consider RC_1 (the signal sequence between CP_3 and CP_4) as the repetition pattern (R_{pat}) and save in a variable called $distanceDTW_{pat}$ the smallest normalized distance between RC_1 and RC_2 or RC_3 . If there is no match among the 3 repetition candidates, then the algorithm will do the following:

- (1) reject the candidate peak CP_3
- (2) define a new searching zone SZ of 10 seconds starting from the candidate peak CP_4
- (3) repeat the pattern searching process in the new SZ

This new searching zone will include one or more candidate peaks (P_4 , P_5 , P_6 , P_7 , and maybe P_8). The pattern detection phase stops when the algorithm finds a repetition candidate that matches 2 or more other RC inside the SZ. This repetition candidate is then considered as the repetition pattern R_{pat} of the current exercise set and passed to the next phase of the similarity match.



(a) Repetition pattern found in the initial searching zone



(b) Repetition pattern found in the self-generated searching zone

FIGURE 10: Examples of exercise repetition patterns found in different searching zones. The blue triangles represent all the candidate peaks before running the algorithm. The green circles indicate the accepted candidate peaks from Step 2.

Figure 10(a) shows an example of a pattern repetition found in the initial searching zone (red color), and Figure 10(b) is another example where the pattern repetition is found in the self-generated searching zone (orange color).

6.3.2. *Similarity Matching and Repetition Count (Phase 2)*. In this phase, we utilize the DTW algorithm to compare each repetition candidate to the exercise repetition pattern R_{pat} from the beginning of the exercise and we count repetition

candidates that are most similar to R_{pat} as actual repetitions. To decide whether two repetitions are similar or not, we need to define a matching threshold parameter. This threshold parameter, which we call β , specifies the maximum allowable Euclidean deviation between two signal sequences to be considered as a match. The higher the value of β , the more divergent (or dissimilar) sequences are considered as a match and the algorithm overperformed. Therefore, a reasonable value for β has to be determined for optimum performance.

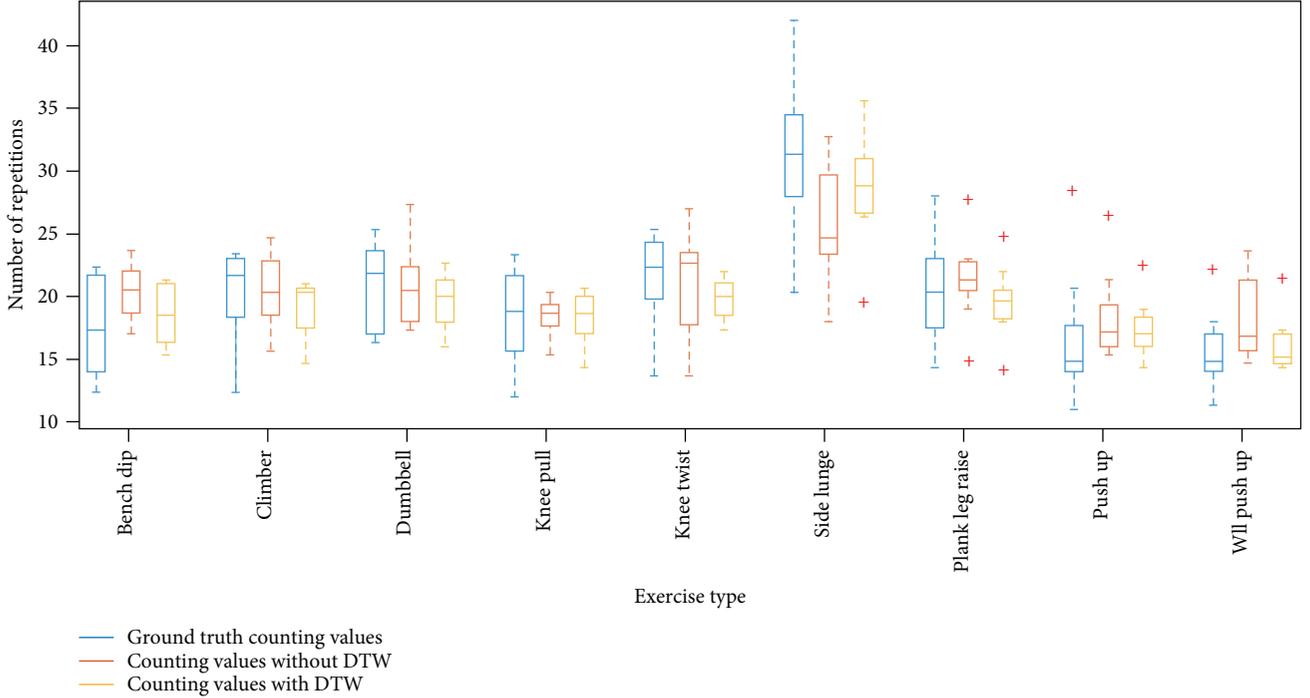


FIGURE 11: Average exercise repetition count from the ground truth and count without and with the DTW.

Through a trial and error approach, we set $\beta = 1.5$ in our current implementation. With this threshold parameter, we compute the threshold DTW (`thresh_dtw`) value which is used to accept or reject a repetition candidate. The `thresh_dtw` is given by

$$\text{thresh_dtw} = \beta * \text{distanceDTW}_{\text{pat}}. \quad (2)$$

If the normalized DTW distance between a repetition candidate and the exercise repetition pattern is less or equal to the `thresh_dtw`, the repetition candidate RC is considered as similar to the exercise pattern R_{pat} and counted as an actual exercise repetition, else the repetition candidate is rejected:

$$RC_j = \begin{cases} \text{accept,} & \text{if } \text{dtw}(RC_j, RC_{\text{pat}}) \leq \text{thresh_dtw,} \\ \text{reject,} & \text{if } \text{dtw}(RC_j, RC_{\text{pat}}) > \text{thresh_dtw.} \end{cases} \quad (3)$$

At the end of this evaluation condition, we sum up the number of similar RC and output the total number of repetitions.

6.4. Counting Algorithm Evaluation and Result. To evaluate the efficiency of our repetition counting algorithm, we utilized the set of data collected during the experiment described in Section 4. We also used the videos recorded during this experiment as ground truth (GT) data.

For each exercise set, we used the algorithm to compute the number of repetitions for each participant and compared this number with the actual number of repetitions obtained from the recorded videos. Out of the 10 exercises, one exercise (pilate arms) was a noncounting exercise. Therefore, we evaluated the counting algorithm over 9 fitness exercises.

Indeed, during the pilate arm exercise, the athlete has to keep the same position during the entire exercise duration without doing any repetitions. We must also note that due to our camera's battery and memory shortage, we missed the following 4 ground truth videos:

- (i) video of P3 performing the knee-twist-in exercise
- (ii) video of P4 performing the plank leg raise exercise
- (iii) video of P7 performing the climber exercise
- (iv) video of P10 performing the climber exercise

We omitted in our evaluation the results obtained from the algorithm for these 4 cases.

To investigate the impact of the DTW in our counting algorithm, we evaluated the algorithm at the end of Step 2 (without DTW) and Step 3 (with DTW) separately.

6.4.1. Results after Step 2 (without DTW). The results described below are obtained after Step 1 and Step 2. The algorithm counted an overall number of 5280 repetitions for all the exercise sets, whereas the ground truth data showed a total of 5186 repetitions. In general, the algorithm slightly overcounted the number of repetitions.

The box plot in Figure 11 depicts the average number of repetition from the ground truth (blue color) and the algorithm without using DTW (red color). The x -axis shows the fitness activities, and the y -axis displays the number of repetitions. We can observe that only 2 exercises (knee-twist-in and side lunge) were undercounted.

Furthermore, for each participant, we computed the average repetition count per exercise (remember that each

TABLE 4: Average mis_count of the algorithm after Step 2 and Step 3 for each exercise and participant.

Exercise types		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Average
Bench dip	Without DTW	-8.0	+1.7	-2.7	+1.0	-1.7	-3.3	-3.7	-1.3	-5.7	-7.7	-3.1
	With DTW	-2.3	+0.3	-1.0	+0.7	+0.7	-1.7	-1.0	+1.0	-3.0	-3.0	-0.9
Climber	Without DTW	0.0	+1.0	-1.7	2.7	-1.3	-0.7	NaN	+1.3	-3.3	NaN	-0.3
	With DTW	+0.7	+2.7	+0.7	+2.0	+2.3	+2.0	NaN	+1.0	-2.3	NaN	+1.1
Dumbbell	Without DTW	+0.7	+4.0	+6.3	-3.3	-1.7	-3.7	+2.3	+0.7	-1.0	-2.0	+0.2
	With DTW	+0.7	+2.7	+4.3	-1.0	-0.7	+1.0	+2.7	+1.7	+0.3	-0.3	+1.1
Knee-pull-in	Without DTW	+0.3	-2.0	-0.3	+2.0	+0.3	+4.7	+1.3	+2.3	-6.3	+1.0	+0.3
	With DTW	+0.7	-1.0	-1.7	+1.3	+0.7	+2.7	+0.0	+2.7	-2.3	+0.3	+0.3
Knee-twist-in	Without DTW	-3.3	-3.7	NaN	+8.3	+1.7	+2.0	+8.0	+1.7	-10.3	+1.7	+0.7
	With DTW	-0.7	+1.3	NaN	+4.7	+3.7	+4.0	+3.0	+1.3	-3.7	+2.0	+1.7
Side lunge	Without DTW	+2.3	-0.7	+9.8	+8.0	-5.7	+0.7	+17.3	+16.7	+9.7	-1.3	+5.7
	With DTW	+1.0	+1.0	+4.5	+3.3	-2.3	+4.0	+6.3	+6.0	+4.0	-0.3	+2.8
Plank leg raise	Without DTW	+1.0	-1.3	-1.0	NaN	-1.7	-3.0	+5.0	-0.3	-5.7	-0.7	-0.9
	With DTW	+2.0	-0.3	+0.7	NaN	+4.0	+0.3	+3.3	+0.3	-3.0	+1.0	+0.9
Push-up	Without DTW	-5.3	-1.3	-2.3	-1.7	+4.7	-6.7	-2.0	+2.0	-2.7	-3.0	-1.8
	With DTW	-3.3	-1.0	-1.0	-0.7	+3.7	-4.3	-3.0	+6.0	-3.3	-2.0	-0.9
Wall push-up	Without DTW	-9.7	+0.7	-0.3	-1.0	-0.3	-9.0	-0.7	+0.7	-5.0	+0.3	-2.4
	With DTW	-1.7	+0.7	+0.3	-0.7	+0.3	-2.3	-0.6	+0.7	-3.0	0.0	-0.6

The high overcounting and undercounting repetition numbers are presented in bold.

exercise has been performed 3 times), from the ground truth (GT) and the algorithm (Alg). Then, we defined and calculated the mis_count values for each exercise, using equation (4). The parameter mis_count allows us to rapidly understand the performance of the algorithm after each step (Step 2 and Step 3).

$$\text{mis}_{\text{count}} = \text{GT} - \text{Alg}. \quad (4)$$

Table 4 gives for each participant the results of the mis_count per exercise. From this table, we observe that without DTW, the proposed algorithm produced high overcounting of +5 up to +17 the actual number of repetitions (e.g., dumbbell of P3, side lunge of P7). We also notice some high undercounting, especially with the knee-twist-in exercise with -10 of the exact repetition number, in the worst case. We defined as a “high overcounting” any repetitions count number obtained from the algorithm that is higher than +4 of the number of actual repetitions. The same definition is applied to the undercount cases, where the “high undercounting” starts from -4 of the number of actual repetitions. These high overcounting and high undercounting results attest that after Step 2, the algorithm is still sensitive to timing and speed variations. The high overcounting and undercounting repetition numbers are highlighted in italic and bold text, respectively, in Table 4.

Although the peak count and time-elapsd approach can count well without looking into the pattern, this approach only is not sufficient for counting exercise repetitions. One way to improve it is to utilize the DTW algorithm to find the repetition pattern of each exercise set and then use this pattern to count the exercise repetitions.

6.4.2. Results after Step 3 (with DTW). At the end of Step 3, the algorithm counts 5053 total repetitions, whereas the ground truth showed a total of 5186 repetitions. The average repetition count obtained after using DTW is almost equal to the number of repetitions observed in the ground truth data. The box plot in Figure 11 depicts the average number of repetitions from the ground truth (blue color) and the algorithm after integrating DTW (orange color).

Table 4 also provides for each participant the result of the mis_count per exercise after Step 3. The highest overcount was about +6.3 of the actual repetition number (e.g., side lunge of P7), whereas the highest observed undercount was about -4.3 of the actual number of repetitions (e.g., push-up of P6).

The difference in the results obtained before and after integrating the DTW algorithm shows that the DTW makes our counting algorithm more robust and accurate.

6.4.3. Overall Results. In evaluating our algorithm, we computed another parameter: error rate in percent. The error rate is calculated as the absolute value of the mis_count over the actual count (GT).

For each exercise and participant, we calculated the error rate by using the average repetition count obtained after integrating the DTW (Step 3). Figure 12(a) plots the average counting error rate of each activity, while Figure 12(b) shows the average counting error rate of each participant.

Independently of the participant and exercise type, there is no result of an error rate higher than 20%. With most participants, the push-up exercise has the highest error rate of 17.70%. This is explicable since the push-up exercise is the most exhausting exercise to be performed over 30 sec. Some participants pause and resume while in the middle of the

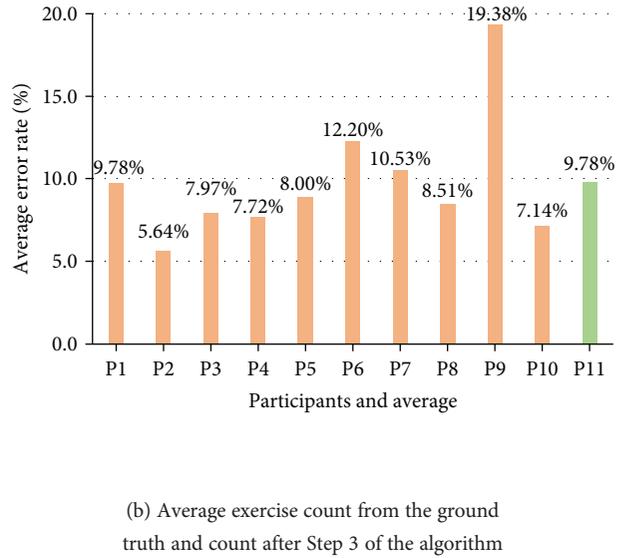
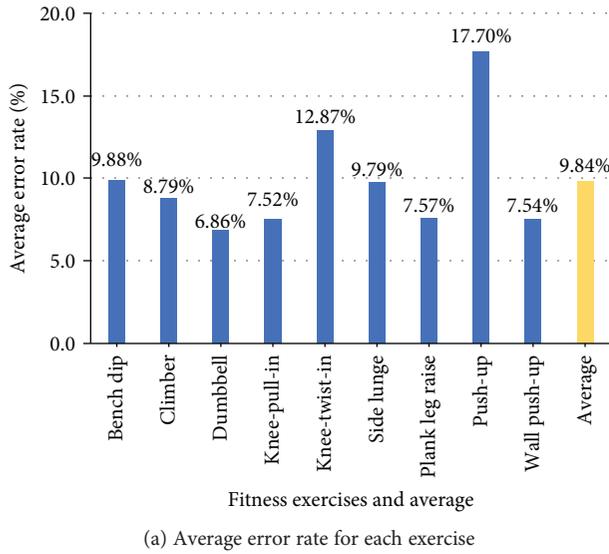


FIGURE 12: Average error rate by exercise and by participant.

push-up exercise, before the time out. Two participants performed the “lady push-up” style because they felt incapable of doing the standard push-up for 30 sec. Two other participants performed the push-up in a different style, which consists of making a subrepetition movement after each normal repetition. This style (a.k.a. “military push-up”) is generally performed by nonamateurs to boost up the effect of the standard push-up.

Only two exercises (knee-twist-in and push-up) have their error rate higher than 10%. The average counting error rate for all exercises is 9.84% and the participant average counting error rate is 9.78%. The worst error rate by participants was obtained with P9 (19.38%), followed by P6 (12.26%). These high error rates might be related to the fact that P9 and P6 did not practice any sports activity in the last six months before the experiment day.

Overall, the results indicate that our repetition counting approach is an acceptable way to make a smart glove that can also count exercise repetitions. The system might work better with regular gym-goers than irregular practitioners. However, we need more testing with experimented fitness practitioners to confirm or deny this hypothesis.

7. Discussion

7.1. User Feedback. In our short exit survey, participants were asked to assess the comfort level of the smart glove on a scale from 0 to 10, with 0 being not comfortable and 10 being comfortable. Figure 13 shows the result of this comfort level assessment. 4 participants out of 10 rated the smart glove with 5 stars, only 1 participant gave a 10-star score, and the others rated it with 6 or 7 stars. The average obtained comfort level of the smart glove is 6.2 over 10.

The second question of the survey (do you have any feedback, comments, suggestions, or ideas related to the device and the system?) revealed that participants want the smart glove to be adjustable, for fitting their hand size. In fact, our current prototype was designed to be “gender neutral with

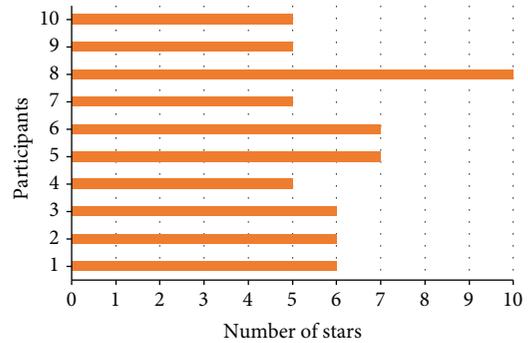


FIGURE 13: Score of the 10-star scale comfort level for each participant.

a unique size.” Therefore, some participants felt like the smart glove was slightly big, while others said that it was small. For example, P1 said, “It would be nice if the width could be adjustable to fit it (the glove) to my arm,” whereas P9 reported, “It (the glove) fits well on my hand but sometimes slides (during the workout).” Another participant (P3) mentioned, “I want to have better tight on my fingers.” From this feedback, we agree that our next prototype should follow the fabrication designs of ordinary fitness gloves already available on the market, in terms of materials, sizes, and appearances.

7.2. Number of FSR. As mentioned in Section 3, we intentionally utilized many FSR sensors (16 FSR) in our first prototype to ensure full coverage and also for investigating the main pressure areas in the palm during fitness activity. Through this investigation, we aim to reduce the total number of FSR sensors in our final smart glove prototype.

By analyzing the data collected during the experiment, it appears that some sensors can be removed without drastically altering the activity recognition and exercise counting results.

TABLE 5: Average sensor values for each exercise.

Sensors position	Bench dip	Climb.	Dumb.	Knee-pull-in	Knee-twist-in	Side lunge	Pilate dip	Plank leg raise	Push-up	Wall push-up
1	54.28	240.54	6.62	91.89	230.71	201.88	219.04	343.82	275.98	56.13
2	231.81	511.77	2.10	227.78	337.49	318.30	248.62	633.29	467.15	71.70
3	341.95	718.57	3.35	331.44	732.16	648.41	745.87	724.94	708.25	226.36
4	274.48	595.46	9.31	381.60	641.78	580.58	673.91	552.13	644.16	267.11
5	50.39	4.72	81.48	0.41	6.00	1.89	0.54	17.66	18.95	0.00
6	244.08	198.89	34.53	34.99	132.48	121.25	151.06	158.61	124.59	1.71
7	478.40	365.18	8.62	81.24	280.89	262.61	314.31	332.83	301.13	8.10
8	540.44	405.72	1.19	135.53	365.07	379.80	373.25	348.93	369.32	71.27
9	0.28	0.28	0.06	1.80	0.53	0.62	0.03	3.06	1.27	0.00
10	21.33	63.72	5.56	13.78	21.98	54.41	32.46	59.86	24.44	1.77
11	29.70	52.35	19.78	16.10	22.57	70.18	31.49	28.70	18.70	0.04
12	7.05	22.94	0.54	21.07	28.75	46.87	36.86	8.28	1.04	0.00
13	4.89	29.09	150.03	2.62	35.28	36.05	28.96	21.48	39.90	0.18
14	66.69	87.37	116.34	25.93	77.09	74.55	54.34	89.54	102.87	7.30
15	1.37	3.81	201.49	1.13	51.67	20.29	6.17	22.33	45.05	1.03
16	3.12	82.12	4.16	9.42	76.11	106.57	51.42	71.28	76.45	7.56
Mean	146.89	211.41	40.32	86.05	190.03	182.77	185.52	213.55	201.20	45.02
Mean/10	14.89	21.14	4.03	8.60	19.00	18.27	18.55	21.35	20.12	4.50

For each activity, we got 30 exercise sets (10 participants \times 3 sets per activity). From these 30 sets per activity, we computed the average force applied to each sensor, to determine whether or not a sensor is activated (used) during that particular exercise. For this evaluation, we used the absolute force applied to the sensors, without normalization.

Table 5 shows the average force values applied to each sensor per exercise, and Figure 14 presents the relation between each exercise and the activation status of the sensors. For each exercise, we also computed the mean of the total forces M_{TF} applied to all sensors (mean of each column in Table 5). If for a given exercise, the average value of a sensor is

- (i) higher than or equal to M_{TF} , we consider the sensor as “always” activated during the exercise (green circle in Figure 14)
- (ii) lower than M_{TF} and higher than $M_{TF}/10$, then the sensor is considered as “sometimes” activated (orange triangle)
- (iii) less than $M_{TF}/10$, then the sensor is considered as “not actually” activated during the exercise (red cross)

The intuition behind this reasoning is that if a value is 10 times less than the average of all values, then this value is significantly low and negligible. From the results of this analysis, we can suggest a new design of the proposed smart glove, without the sensors at position 5 and 9. Indeed, at these positions, the sensor values are negligible for most exercises. The sensors on the fingers (positions 13, 14, 15, and 16) can also be removed or slightly moved down because they are

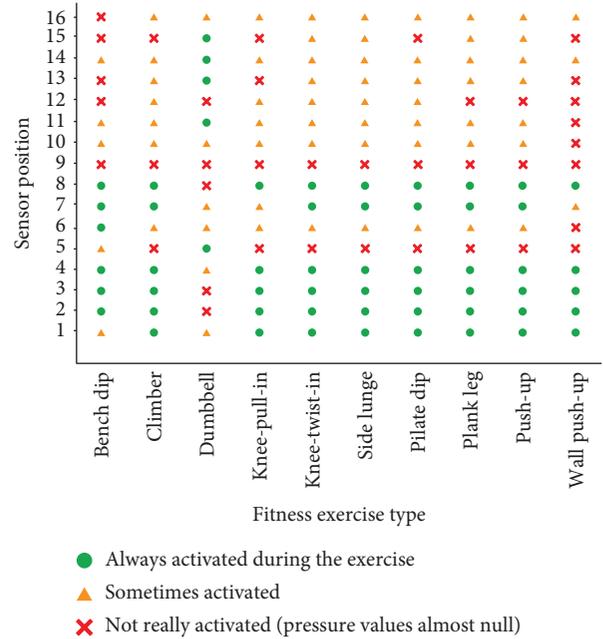


FIGURE 14: Sensor activation status during fitness exercise.

frequently inactivated or activated only a few times for most exercises except for the dumbbell curl exercise. Sensors on positions 1, 2, 3, 4, 7, and 8, which are generally highly activated, should remain in the same spots for the next prototype.

7.3. Fitness Exercises without Direct Pressure on the Palm. One limitation of our current system is the tracking of exercises such as running that does not produce pressure

variation in the palm. To overcome this issue, we intend to integrate inertial sensors such as accelerometers or gyroscopes into the DSU (data sampling and communication unit) of our next smart glove prototype. By adding such inertial sensors into our system, we will be able to track pressureless exercises, as well as improve the exercise recognition and counting performance of the proposed system.

7.4. Number of Participants. Another limitation of the present work was the low number of participants during the experiment, which meant we could not determine whether the obtained results of the current prototype are statistically significant. However, these results confirmed our intuition and idea that it is possible to use biomechanical data from the hand palms to monitor fitness activity. Moreover, the current experimental results provide valuable information and orientation that could be employed to improve the overall system performance. Notwithstanding the above, in our future work, we intend to validate the proposed methods with a larger number of participants.

8. Conclusion

In this work, we investigated a novel smart glove-based system for tracking indoor fitness activity. Our approach exploits the interactions between the hand palm and the working environment to assess fitness activities. The system integrates 16 FSR sensors into a fitness glove to identify fitness activities and count the repetition of an exercise, by analyzing a time series of the pressure distribution applied to the hand palm during the exercise.

We presented the design of the smart glove and evaluated the exercise recognition performance and the accuracy of the repetition counting algorithm of the system. Our validation experiment with 10 healthy participants over 10 common fitness exercises showed an overall exercise recognition accuracy of 88.00% for the person-dependent evaluation and 82.00% for the person-independent evaluation. The evaluation of the repetition counting algorithm achieved an average counting error rate of 9.85%. Based on our results, we concluded that a smart glove that collects and analyzes hand palm pressure could be used to track and assess fitness activities.

Clearly, in its current design, the system alone is not enough for tracking exercises that do not generate pressure in the hand palms. However, it provides valuable results that can be combined with the data from inertial sensors, to develop more complete systems for supporting fitness practitioners.

Data Availability

The smart glove data (in .csv format) used to support the findings of this study have been deposited in the “FSR-Smart-glove-data” public repository on Github and are accessible via the following link: <https://github.com/edess/FSR-Smart-glove-data>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partly supported by JST PRESTO and JSPS KAKENHI Grant Number JP16H01721.

References

- [1] F. J. Penedo and J. R. Dahn, “Exercise and well-being: a review of mental and physical health benefits associated with physical activity,” *Current Opinion in Psychiatry*, vol. 18, no. 2, pp. 189–193, 2005.
- [2] M. Reiner, C. Niermann, D. Jekauc, and A. Woll, “Long-term health benefits of physical activity—a systematic review of longitudinal studies,” *BMC Public Health*, vol. 13, no. 1, p. 813, 2013.
- [3] D. E. Warburton, C. W. Nicol, and S. S. Bredin, “Health benefits of physical activity: the evidence,” *Canadian Medical Association Journal*, vol. 174, no. 6, pp. 801–809, 2006.
- [4] K. F. Janz, J. D. Dawson, and L. T. Mahoney, “Tracking physical fitness and physical activity from childhood to adolescence: the Muscatine study,” *Medicine & Science in Sports & Exercise*, vol. 32, no. 7, pp. 1250–1257, 2000.
- [5] S. Asimakopoulos, G. Asimakopoulos, and F. Spillers, “Motivation and user engagement in fitness tracking: heuristics for mobile healthcare wearables,” *Informatics*, vol. 4, no. 1, p. 5, 2017.
- [6] M. Hassan, F. Daiber, F. Wiehr, F. Kosmalla, and A. Kruger, “FootStriker: an EMS-based foot strike assistant for running,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 1, pp. 1–18, 2017.
- [7] M. M. Jensen and F. Mueller, “Running with technology: where are we heading?,” in *OzCHI '14 Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: the Future of Design*, pp. 527–530, Sydney, New South Wales, Australia, December 2014.
- [8] D. Morris, T. S. Saponas, A. Guillory, and I. Kelner, “RecoFit: using a wearable sensor to find, recognize, and count repetitive exercises,” in *CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3225–3234, Toronto, Ontario, Canada, May 2014.
- [9] D. F. Whelan, M. A. O'Reilly, T. E. Ward, E. Delahunt, and B. Caulfield, “Technology in rehabilitation: evaluating the single leg squat exercise with wearable inertial measurement units,” *Methods of Information in Medicine*, vol. 56, no. 02, pp. 88–94, 2018.
- [10] M. Sundholm, J. Cheng, B. Zhou, A. Sethi, and P. Lukowicz, “Smart-mat: recognizing and counting gym exercises with low-cost resistive pressure sensing matrix,” in *UbiComp '14 Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 373–382, Seattle, WA, USA, September 2014.
- [11] LiveStrong, “The benefits of wearing weight lifting gloves,” 2017, <https://www.livestrong.com/article/17819-benefitswearing-weight-lifting-gloves/>.
- [12] K.-H. Chang, M. Chen, and J. Canny, “Tracking free-weight exercises,” in *UbiComp 2007: Ubiquitous Computing. UbiComp 2007. Lecture Notes in Computer Science*, vol. 4717, J.

- Krumm, G. D. Abowd, A. Seneviratne, and T. Strang, Eds., pp. 19–37, Springer, Berlin, Heidelberg, 2007.
- [13] I. Pernek, K. A. Hummel, and P. Kokol, “Exercise repetition detection for resistance training based on smartphones,” *Personal and Ubiquitous Computing*, vol. 17, no. 4, pp. 771–782, 2013.
- [14] Q. Ye, M. Seyedi, Z. Cai, and D. T. H. Lai, “Force-sensing glove system for measurement of hand forces during motorbike riding,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, Article ID 545643, 2015.
- [15] B. Zhou, M. Sundholm, J. Cheng, H. Cruz, and P. Lukowicz, “Never skip leg day: a novel wearable approach to monitoring gym leg exercises,” in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–9, Sydney, NSW, Australia, March 2016.
- [16] FH page, “Fitbit official site for activity trackers and more,” 2018, <https://www.fitbit.com/home>.
- [17] GW Technology, “Wearables and smartwatches Garmin,” 2018, <https://buy.garmin.com/enUS/US/wearabletech/wearables/c10001-c10002-p1.html>.
- [18] Fitbit Community, “Logging strength training,” 2017, <https://community.fitbit.com/t5/Alta/Logging-strength-training/tdp/1350809>.
- [19] ActiveLinxx, “Power your FitLinxx system,” 2016, <https://www.activelinxx.com/services/>.
- [20] Google Fit Help, “Fit isn’t tracking activities correctly,” 2018, <https://support.google.com/fit/answer/>.
- [21] B. Zhou, H. Koerger, M. Wirth et al., “Smart soccer shoe: monitoring football interaction with shoe integrated textile pressure sensor matrix,” in *ISWC '16 Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pp. 64–71, Heidelberg, Germany, September 2016.
- [22] E. A. Akpa, M. Fujiwara, Y. Arakawa, H. Suwa, and K. Yasumoto, “Gift: glove for indoor fitness tracking system,” in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 52–57, Athens, Greece, March 2018.
- [23] A. Sadun, J. Jalani, and J. Sukor, “Force sensing resistor (FSR): a brief overview and the low-cost sensor for active compliance control,” in *Proceedings Volume 10011, First International Workshop on Pattern Recognition*, Tokyo, Japan, July 2016.
- [24] Interlink Electronics, “FSR 400 series data sheet,” 2016, <http://interlinkelectronics.com/datasheets/DatasheetFSR.pdf>.
- [25] Tekscan, “FlexiForce A201 datasheet,” 2016, <https://www.tekscan.com/resources/product/flexiforce-a201-datasheet>.
- [26] AdaFruit, “Adafruit feather 32u4 bluefruit LE,” 2016, <https://learn.adafruit.com/adafruit-feather-32u4bluefruit-le/overview>.
- [27] Physical Activity Council, *2018 Participation Report: the Physical Activity Council’s Annual Study Tracking Sports, Fitness, and Recreation Participation in the US*, 2018, <http://www.physicalactivitycouncil.com/>.
- [28] BodyBuilding, “All plans - find a workout plan,” 2016, <https://www.bodybuilding.com/workout-plans>.
- [29] E. T. Luhanga, A. A. E. Hippocrate, H. Suwa, Y. Arakawa, and K. Yasumoto, “Identifying and evaluating user requirements for smartphone group fitness applications,” *IEEE Access*, vol. 6, pp. 3256–3269, 2018.
- [30] T. Huynh and B. Schiele, “Analyzing features for activity recognition,” in *sOc-EUSAI '05 Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pp. 159–163, Grenoble, France, October 2005.
- [31] M. Zeng, L. T. Nguyen, B. Yu et al., “Convolutional neural networks for human activity recognition using mobile sensors,” in *6th International Conference on Mobile Computing, Applications and Services*, pp. 197–205, Austin, TX, USA, November 2014.
- [32] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Computing Surveys*, vol. 46, no. 3, pp. 1–33, 2014.
- [33] L. Rokach, “Ensemble-based classifiers,” *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.
- [34] A. Savitzky and M. J. E. Golay, “Smoothing and differentiation of data by simplified least squares procedures,” *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.
- [35] J. Parkka, M. Ermes, P. Korpiainen, J. Mantyjarvi, J. Peltola, and I. Korhonen, “Activity classification using realistic data from wearable sensors,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 119–128, 2006.
- [36] L. Muda, M. Begam, and I. Elamvazuthi, “Voice recognition algorithms using Mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques,” 2010, <http://arxiv.org/abs/1003.4083>.
- [37] J. Blackburn and E. Ribeiro, “Human motion recognition using isomap and dynamic time warping,” in *Human Motion – Understanding, Modeling, Capture and Animation. HuMo 2007. Lecture Notes in Computer Science, vol 4814* pp. 285–298, Springer, Berlin, Heidelberg.
- [38] J. Aach and G. M. Church, “Aligning gene expression time series with time warping algorithms,” *Bioinformatics*, vol. 17, no. 6, pp. 495–508, 2001.



Hindawi

Submit your manuscripts at
www.hindawi.com

