

A Low-Cost Smart Glove System for Real-Time Fitness Coaching

Yongpan Zou¹, *Member, IEEE*, Dan Wang, Shicong Hong, Rukhsana Ruby², *Member, IEEE*,
Dian Zhang, *Member, IEEE*, and Kaishun Wu³, *Member, IEEE*

Abstract—Strength training is becoming increasingly popular among all age groups, as it helps the participants increase muscle strength, improve body flexibility, reduce health risks, and reshape physical forms. However, strength training imposes strict regulations on gestures and requires professional instruction in real time for the sake of body-building efficiency and safety. For this purpose, in this article, we propose a novel low-cost system named iCoach, to provide real-time monitoring and coaching service for strength training participants. Specifically, we design and implement a smart fitness glove, which can be seamlessly equipped with a pervasive inertial unit. With this customized but low-cost device, we can recognize various training programs, detect nonstandard behaviors while exercising, and assess exercising qualities of a user. Our primary experimental results show that iCoach can recognize 15 sets of training programs, detect three common nonstandard behaviors, and assess the quality of training with high accuracy and reliability.

Index Terms—Inertial measurement unit (IMU), strength training, training exercise assessment.

I. INTRODUCTION

NOWADAYS, individuals pay more attention to physical health and body shape. Besides nutrition, strength training appeals a mushrooming number of people across all age groups, especially the youngsters. The underlying reason is that strength training can bring about multiple benefits all at once, such as increasing muscle strength, improving body flexibility, reducing health risks, and reshaping physical form [1]. However, differing from other fitness activities like aerobic workouts, strength training puts stricter requirements on postures and demands more professional instruction, in order to achieve better fit-keeping effect and avoid muscle injuries. Compared with those activities, posture, speed, training load,

and sets play critical roles in guaranteeing training effect and avoiding body injuries. Without timely feedback, even minute nonstandard behaviors in strength training activities may cause injuries in an instant or lead to the deformation of the body shape in the long run.

Consequently, in order to achieve a better workout effect, present fitness participants tend to employ professional fitness coaches or turn to fitness APPs [2], [3] as a compromise. Nevertheless, hiring such a personal coach is rather expensive and unbearable for general bodybuilders. Moreover, existing APPs only provide fitness tutorials and post-recording service through the user's typing. They do not monitor the training process of the participants in real time, and hence correcting the nonstandard behaviors to avoid muscle injuries is not supported as well.

Although there are already many works in analyzing aerobic exercise and general activities [4]–[8], they cannot be applied to strength training activities owing to their uniqueness. Some researchers propose to use an RFID system to monitor free-weight exercise [9], [10]. However, it requires instrumented fitness equipment (dumbbells, barbells, and machines) with numerous tags and readers in order to cover a gym entirely. From the point of hardware cost, this is not a favorable manner since even a commercial RFID reader costs hundred times more compared to the pervasive sensors. Even worse, complex layout, human movements, and RF interference in the gym can easily interfere with the propagation of wireless signals and degrade the performance of tracking exercise. The most similar works that use inertial sensors (e.g., accelerometer and/or gyroscope) to monitor free-weight exercise appeared in [11] and [12]. However, both of these merely focus on detection, recognition, and counting of repetitive activities, which can only provide coarse information about activities. Furthermore, these works do not detect nonstandard components and hence do not provide a comprehensive assessment of exercise activities.

In contrast, we design and implement iCoach, a novel intelligent fitness glove as shown in Fig. 1, which is designed for monitoring, coaching, and assessing strength training exercises in a more comprehensive manner. More specifically, iCoach is a smart fitness glove with a commercial inertial measurement unit (IMU), including an accelerometer, a gyroscope, and a magnetometer embedded in its wrist band. In our design, iCoach works by utilizing IMU data to recognize exercise activities, detect nonstandard behaviors, and assess workout quality. Different from the previous systems, iCoach

Manuscript received December 18, 2019; revised February 28, 2020 and March 19, 2020; accepted March 20, 2020. Date of publication March 24, 2020; date of current version August 12, 2020. This work was supported in part by China NSFC under Grant 61802264, Grant 61872248, and Grant U1736207; in part by the Shenzhen Science and Technology Foundation under Grant JCYJ20180305124807337 and Grant ZDSYS20190902092853047; in part by Guangdong NSF under Grant 2017A030312008; in part by the Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China under Grant 161064; in part by GDUPS (2015); and in part by the Natural Science Foundation of SZU under Grant 860-000002110537 and Grant 827-000175. (Corresponding author: Kaishun Wu.)

Yongpan Zou, Dan Wang, Shicong Hong, Rukhsana Ruby, and Kaishun Wu are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: yongpan@szu.edu.cn; wangdan2018@email.szu.edu.cn; ruby@szu.edu.cn; wu@szu.edu.cn).

Dian Zhang is with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong (e-mail: zhangdian@ln.edu.hk).

Digital Object Identifier 10.1109/IJOT.2020.2983124



Fig. 1. Left: The present prototype of iCoach. Right: Experimental scenarios. iCoach is a smart device and application that monitors exercise activities and provides feedback in real time. It can be seamlessly embedded with fitness glove and thus is nearly unobtrusively. We conduct experiments in a university gym to evaluate its performance.

is a more comprehensive monitoring, coaching, and interactive system, with which users can not only obtain some basic information but can further acquire more critical details, such as nonstandard behaviors and assessment of their exercising process. Moreover, iCoach can act as a real-time interactive interface for trainers. If any abnormal behaviors during the course of exercise are detected, corresponding instructions are sent to the users for correcting errors and avoiding injuries instantaneously. Our experimental results show that iCoach can identify training activities with a high accuracy (precision: 94.7% and recall: 95.0%). The iCoach can also detect more fine-grained nonstandard behaviors with reasonable accuracy (precision: 90.7% and recall: 90.3%). Finally, we also demonstrate the reliability of assessing the quality of doing exercise via iCoach.

To summarize, we highlight our main contributions in this article as follows. First, we demonstrate the feasibility and reliability of monitoring, assessing, and coaching strength training exercise using low-cost sensors in an unobtrusive manner. We do not require instrumenting the environment with any additional sensors, and can easily embed sensors in fitness gloves. Second, we design and implement a real-time system, iCoach, and conduct experiments in practical settings to evaluate its performance. The results show that iCoach achieves an accuracy higher than 95% and has a less than 90-ms response latency. Compared with professional coach, iCoach achieves satisfactory assessment quality.

II. RELATED WORK

The iCoach is closely related to the human activity recognition concept in terms of the following two categories.

A. Wearable Sensor-Based Activity Recognition

Wearable sensors have been used to recognize various human activities in previous works [11], [13]–[22]. Researchers make use of wrist-mounted inertial sensors to monitor the arm-based activities [11], [13]–[15], [17] or track the arm motions [16]. Among these, the works in [15] and [17] leverage accelerometer on the wrist to detect and

recognize smoking and eating activities, respectively, while ArmTrak [16] tracks one's precise arm movement with the help of inertial sensors in a smartwatch. LookUp [22] uses the shoe-mounted inertial sensors for classifying physical locations based on surface gradient profile and step patterns. MyBehavior [23] makes use of the multimodal input on the smartphone, such as activity, food intake, exercise, etc., to learn a user's physical activity and dietary behavior. The system in [24] uses a single body-worn accelerometer to estimate caloric expenditure. Although all the above works are relevant to activity recognition with inertial sensors, they do not focus on monitoring and assessing strength training exercise, which is a more specific area requiring different data processing techniques. Most similar to this article, the works [13] and [14] focus on recognizing strength training activities with inertial sensors. However, the differences between this article and them lie in two aspects. For one thing, the categories of activities in our work is more than that in other works, but the recognition accuracy remains high. For another thing, we not only design and implement a real-time strength training activities recognition system but also conduct a comprehensive assessment of activities.

B. Radio-Frequency-Based Activity Recognition

Another category of activity recognition works exploit radio-frequency signals (e.g., WiFi signals and RFID signals) to sense human activities [10], [25]–[29]. WiSee [26] exploits the Doppler shift effect of WiFi signals to recognize the human gestures. E-eyes [25] utilizes the channel state information (CSI) of WiFi signals to monitor the daily indoor activities. Similarly, Smokey [27] leverages CSI to detect one's smoking activity. Differently, the work in [28] proposes a quantitative model to estimate the walking speed of a person from CSI values. FEMO [10] provides a free-weight exercise monitoring scheme by attaching RFID tags on the dumbbells and then leveraging the Doppler shift profile of the reflected backscatter signals for activity recognition. The work [29] designs a CSI-based strength training monitoring system with commercial WiFi devices. The difference between our work and these works can be recognized in three aspects. First, except the works [10] and [29], till now, there is almost no research work on strength training. Second, wireless signals are easy to be interfered with by external environmental factors such as movements of other people.

III. TESTED TRAINING ACTIVITIES

Strength training is specialized in terms of resistance to induce muscle contraction and covers rather diverse categories of activities. In this article, we focus on 15 sets of training programs as shown in Fig. 2. We select these training programs since they train different parts of upper body muscles covering a person's major muscle groups. Moreover, these training programs are among the top popular activities according to the fitness training guidebooks [1], [30] and questionnaire investigation. Specifically, the selected activities aim at training brachialis, pectoralis, abdominal muscle, dorsal muscle, and deltoid. In fitness training, a "repetition" is the basic unit of

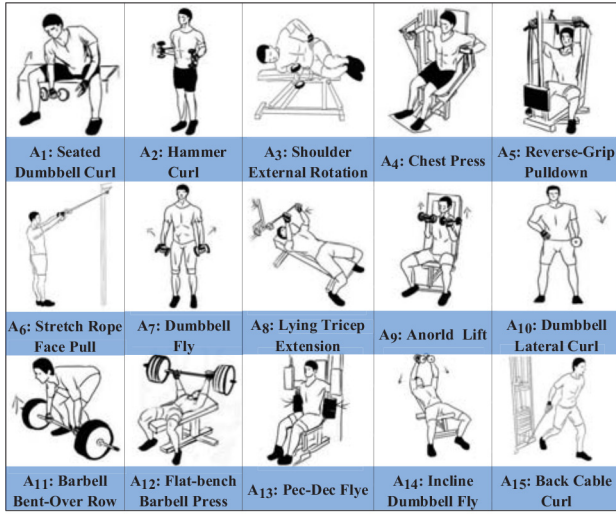


Fig. 2. Tested training programs in order to evaluate the performance of iCoach. We test 15 sets of most common programs, which involve free-weight training exercise and machine workouts. It is noted that iCoach cannot only be applicable for the aforementioned 15 activities. Rather, it can be extended for more upper-body activities.

doing exercise, while a “set” usually contains a certain number of repetitions.

Various nonstandard behaviors may happen during a training process [31]–[33]. As for this, we select three common ones as described in the following.

- 1) *Case 1 (The Overall Speed of a Repetition Is Too Fast or Too Slow)*: The overall speed of a repetition plays a significant role in strength training. A faster movement may lead to stretching and warping muscle in a fierce manner. A slower movement may lead to ineffective muscle training due to the lack of effective resistance.
- 2) *Case 2 (The Speeds of Outward and Backward Processes Are Not Balanced)*: The balance of outward–backward processes is another critical metric of the strength training. The tension of muscle and sufficient control should be kept constant all the time to stimulate muscle for obtaining the benefit of an effective workout. Balanced centripetal contraction (outward) and centrifugal contraction (backward) benefit to enhancing the same powerful bidirectional muscle contraction strength and avoid disequilibrium. In particular, muscle injuries most likely to appear when the centrifugal contraction is too fast.
- 3) *Case 3 (The Repetitions Are Not Stable With Noticeable Shakes)*: Those repetitions with obvious shakes but can be recognized correctly are another kind of nonstandard behavior. On account of the shake of arms, the muscle cannot maintain certain tension that is required for exercise.

IV. SYSTEM DESIGN

In this section, we give a detailed introduction to the specific design of iCoach. Fig. 3 shows the overall workflow diagram of the iCoach system. Its key components include preprocessing, segmentation, activity recognition, and analysis

and assessment methods. In the following parts, we provide more details of the system composition.

A. Preprocessing

In the preprocessing stage, we apply calibration to IMU sensors and denoising operation on raw signals. Specifically, we apply an improved nine-position static calibration method to calculate the bias and scale factor of the IMU sensors in each axis [34]. Later, an exponentially weighted moving average (EWMA) filter, which can be described by (1), is operated on the calibrated data in each axis to remove random noise caused by irrelevant motions

$$a_i^l = a_{i-1}^l + \lambda \times (a_i - a_{i-1}^l), \quad i > 0, \quad a_0^l = a_0 \quad (1)$$

where a_i is the i th calibrated data and a_i^l is the value after passing through the smoothing filter. We set the smoothing factor λ to 0.1 through the empirical optimization on the collected data.

B. Segmentation

To achieve high recognition accuracy and better exercise assessment, it is significant to precisely segment a data stream, i.e., detecting the start point (SP) and end point (EP), to align the data with the actual exercise process. Besides the precision requirement, we also need to take the user diversity into account. That is, different users have different exercise patterns, such as speed and stability, which makes firm threshold-based segmenting utilized in most previous works fail to work well.

To deal with the above challenges, we make use of an observation that there exists a short pause between consecutive exercise repetitions, and propose a double adaptive thresholds-based method to detect the exact SP and EP of each repetition. Although the IMU contains 9-axis data, we only make use of accelerometer and gyroscope measurements since magnetic measurements are easily affected by the surroundings. As for the gyroscope measurements, we calculate the average absolute value of each axis in a window of length w as three indicators, namely, I_{gx} , I_{gy} , and I_{gz} . Then, the 3-axis thresholds of gyroscope data, namely, γ_{gx} , γ_{gy} , and γ_{gz} are calculated by

$$\gamma = \beta \cdot \sigma \quad (2)$$

where $\sigma = [(\sum_{i=1}^n |a_i|)/n]$, which represents the average absolute value of noise signals.

To estimate the value of σ , we collect data for about 1 s once the Bluetooth connection succeeds. During this period, the users are requested to keep their hands still naturally such that the collected data reflect the personalized stability of their arms. This indicates that different users have varied σ values. Thus, each time a user begins doing exercise, σ is adjusted adaptively based on the current stability value of that individual. As for the accelerometer, since the data are affected by gravitational acceleration, to make the activity detection method work for different fitness programs, we use the square magnitude of acceleration (SMA) to estimate the total acceleration. The SMA can be derived using the following equation:

$$\text{SMA}(i) = (ax_i)^2 + (ay_i)^2 + (az_i)^2 \quad (3)$$

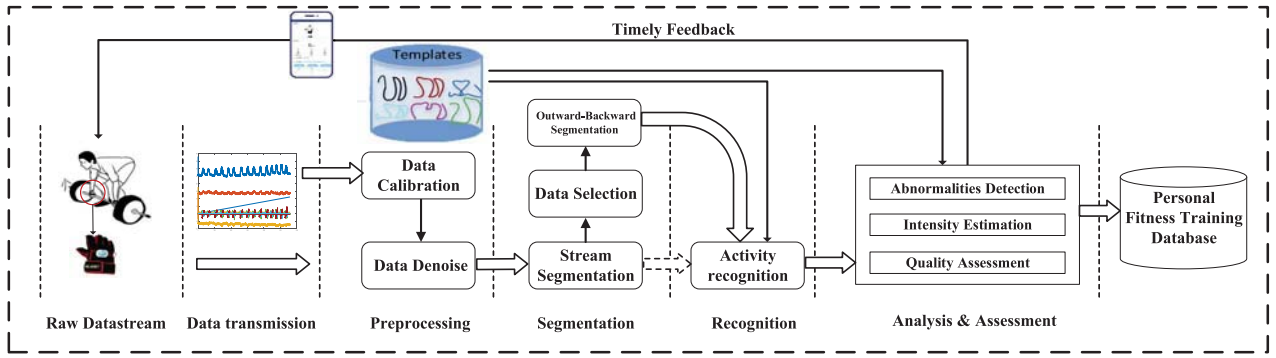


Fig. 3. Work flow of iCoach. The iCoach mainly consists of four main parts, that are for data preprocessing, data segmentation, activity recognition, and analysis and assessment of training activities. Different from the previous works, iCoach stands out principally for its ability to detect nonstandard behaviors with acceptable performance and assessing the quality of training exercise comprehensively.

where ax_i , ay_i , and az_i are the acceleration of the i th sample on the x -, y -, and z -axes, respectively. Similarly, we calculate the variance of the resultant SMA signal in the same window as the acceleration indicator, namely, I_a . Then, the SMA threshold τ is calculated by

$$\tau = \alpha \cdot \delta \quad (4)$$

where $\delta = ([\sum_{i=1}^n (SMA_i - \mu)]/n)$ and $\mu = ([\sum_{i=1}^n SMA_i]/n)$. δ represents the variance of the same noise signals and is adjusted adaptively based on the stability value of each individual. β and α are constant parameters empirically set to 1.5 and 2, respectively, in the iCoach.

Based on the above, we put forward the criterion of determining whether a sliding window is active or inactive. A potential starting window of a repetition is detected if in the current window, any two of the gyroscope indicators I_{gx} , I_{gy} , and I_{gz} exceed γ_{gx} , γ_{gy} , and γ_{gz} defined in (2), or the acceleration indicator I_a exceeds τ defined in (4); otherwise, the corresponding sliding window is detected as an end one. Consequently, the signal segment between starting and ending windows can be extracted as a repetition candidate. However, due to irrelevant motion interference, there exist fakes in the candidates. Thus, if the length of a candidate exceeds a constant threshold Γ , representing the minimum fitness motion length, we consider it to be a true repetition. By this means, the interference of irrelevant movement can be removed. The entire process of activity segmentation is summarized in Algorithm 1. In iCoach, since the sampling rate is 40 Hz, the window length w and the step length p are set to 20 and 5 samples, respectively.

Although there are extensive works in activity recognition with IMU, their methods cannot be applied in our work. As for activity segmenting, we implement two common methods, namely, the peak finding algorithm in [13] and the KL divergence algorithm in [10], and compare them with our approach. The peak finding algorithm segments activities by identifying peaks in data streams. According to [13], it can only extract a repetition when there are no time gaps between consecutive repetitions. The KL divergence algorithm utilizes the probability density distribution to identify active and static windows. However, the window size should be carefully selected for each user, as a small window size tends to segment a repetition

Algorithm 1: Motion Signal Detection Algorithm

Input: Timeseries, X_n ; WindowSize; lapstep;
Output: A start point of an activity, t_{start} ; The corresponding end point, t_{end} ;

```

1 Initialize: state = static;
2 if in a window  $W_t = \langle x_{t-WindowSize+1}, \dots, x_t \rangle$ 
   (state == static) and
   ((( $I_{gx} > \gamma_{gx}$ ) + ( $I_{gy} > \gamma_{gy}$ ) + ( $I_{gz} > \gamma_{gz}$ )) > 1) or  $I_a > \tau$  then
3    $t_{start} = t - \text{lapstep}$ ;
4   state = active;
5   while not (state == active and  $I_{gx} \leq \gamma_{gx}$  and
6      $I_{gy} \leq \gamma_{gy}$  and  $I_{gz} \leq \gamma_{gz}$  and  $I_a \leq \tau$ ) do
7     window slide forward to  $W_{t'}$  with lapstep;
8   end
9    $t_{end} = t' - \text{lapstep}$ ;
10  state = static;
11  return  $t_{start}, t_{end}$ 
12 else
13   window slide forward with lapstep;
14   goto 2;
15 end

```

into massive fragments, while a large window size leads to a higher time delay. In addition, the computational cost of identifying the resting intervals (i.e., static periods) is twice compared to our method with the same window size. Different from the existing complex segmenting schemes, we segment data streams in a simple but effective way based on kinematic interpretation of IMU data. Fig. 4 shows examples of the smoothed measurements (i.e., 3-axis gyroscope) and segmentation result of six randomly selected activities. As we can see, the KL divergence algorithm can detect segments with an acceptable detected accuracy when a larger window length is applied. However, it cannot locate the SPs and EPs of a segment precisely. Moreover, as shown in Fig. 4(a) and (c), a larger window length also makes it easy to merge multiple activities into a single one while the rest intervals between activities are too short. As for the peak finding algorithm, after finding the peaks of a data stream, we have to purify the

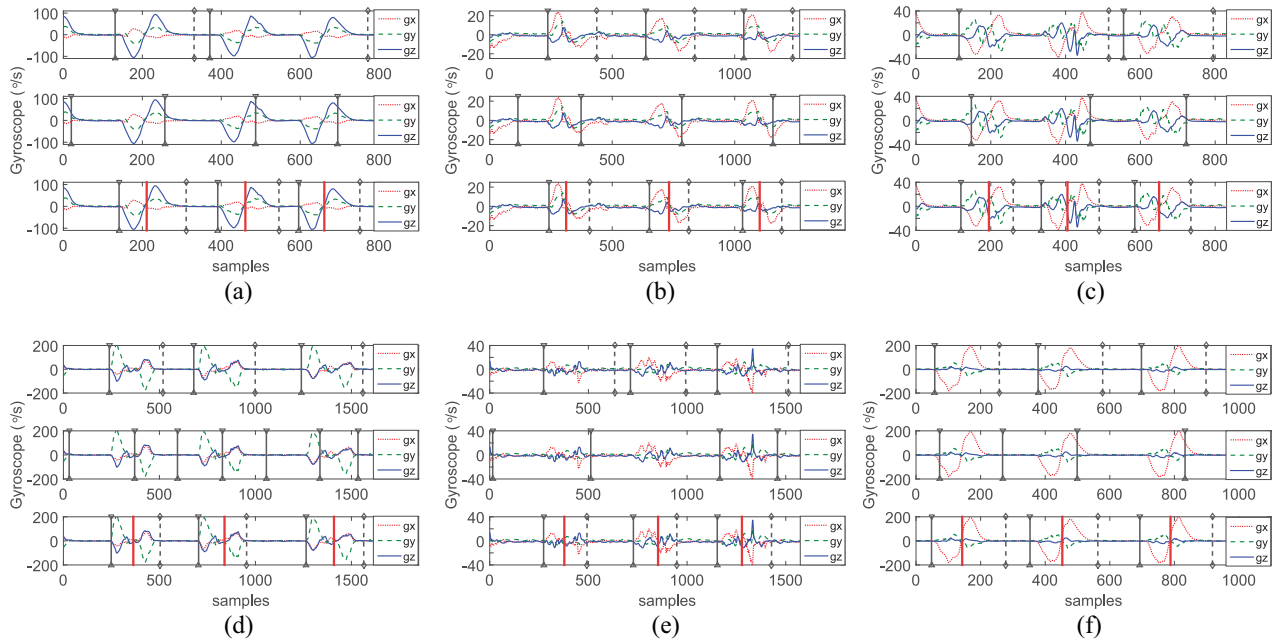


Fig. 4. Examples of filtered 3-axis gyroscope value and segmentation result of activity A_2 , A_4 , A_6 , A_9 , A_{11} , and A_{15} (three repetitions in each subfigure). In each subfigure, the top, middle, and bottom figures show segmentation results of the same signal using different segmentation methods. More specifically, the top figure shows the segmentation result using the KL divergence algorithm in [10], the middle figure shows the result using the peak finding algorithm in [13], and the bottom figure shows the activity segmentation result and outward-backward segmentation result using the algorithms described in Sections IV-B and IV-C, respectively. In the top and bottom figures, the first black vertical solid line and the following black vertical dotted line represent the SP and EP of a detected segment. In addition, the red vertical solid line in the bottom figure represents the transition point (TP) of the detected segment. In the middle figures, the signal between two consecutive black vertical solid lines represents a detected segment.

probable peaks by means of the fake peak removal algorithm in [13]. Since a short rest interval is required between consecutive repetitions, the peak finding algorithm tends to find peaks in the intervals with massive fake peaks. Furthermore, the varying durations of exercise repetitions and rest intervals make it more difficult to segment activities correctly. In addition, the detected segments in [13] consist of both rest intervals and genuine motion signals, which is unbeneficial for the following fine-grained quality assessment. Finally, as shown in Fig. 4, our method can effectively detect SP and EP of each activity and hence the measurements are distinguishable. Due to the limited space, we only display part of sensor measurements (i.e., gx , gy , and gz) of several strength training activities in Fig. 4 for the sake of clear description.

C. Outward-Backward Segmentation and Recognition

After segmentation, we obtain the accelerometer and gyroscope segments of each activity. To recognize activities, we match data segments of an unrecognized activity with labeled standard templates obtained from the coach. Here, we refer data segments of each training activity prestored in the system as “templates.” Likewise, we refer to a set that contains one template for each tested activity as “a templates set.” There are different metrics of measuring the matching degree (i.e., similarity) between two data series, such as the Euclidean distance, cosine similarity, dynamic time warping (DTW), and the like. In our application, we adopt the DTW metric to measure the similarity between two data segments. DTW is a widely used technique for pattern recognition, especially

speech recognition. Compared with other metrics, DTW is able to take the elastic shift operation in the time domain into account and match sequences that are similar but out of phase. As a result, it is a powerful tool in measuring the similarity between two sequences which may vary in length. The smaller the DTW value between two sequences, the more similar they are. Consequently, to recognize an unknown activity, we match its data segments, namely, accelerometer and gyroscope reading segments, with corresponding templates of each activity and choose the activity that outputs the minimum DTW value as the recognized result. However, in spite of excellent recognition accuracy, this method has poor computational efficiency which affects the real-time performance.

Consequently, we started to use the FastDTW technique [35] to substitute the DTW technique. The FastDTW algorithm is a linear and accurate approximation of the DTW technique while the time complexity of the DTW technique is quadratic. Based on the DTW technique, the FastDTW algorithm uses a multilevel approach that recursively projects a warp path from a coarser resolution to the current resolution and refines it. Nevertheless, although we apply the FastDTW algorithm in practice, it has worse performance in terms of both recognition accuracy and computational efficiency.

In order to achieve high recognition accuracy and low computational delay, we devise a novel activity recognition algorithm based on the DTW technique, namely, Half-DTW. The intuition behind the approach is based on two key observations. First, a fitness motion segment (that represents a repetition) can be further divided into two parts. For convenience,

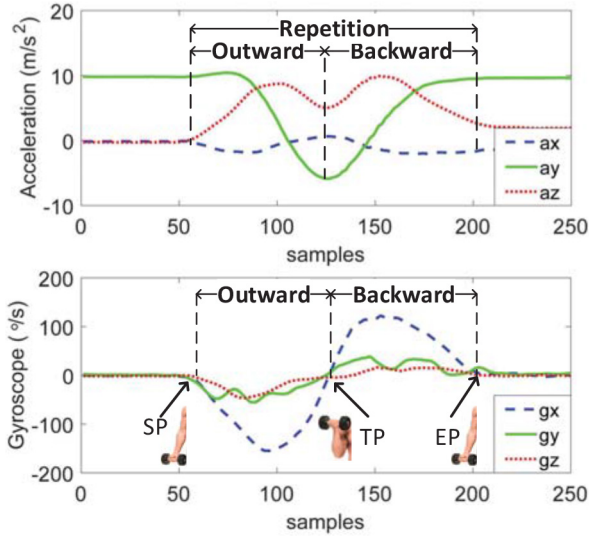


Fig. 5. Sample illustration of the relationship among {SP, TP, EP}, outward-backward parts, and the arm movements in a repetition (smoothed 3-axis acceleration and 3-axis gyroscope of activity A_1 captured by the smart glove are shown in the above and below subfigures, respectively).

we call these two parts as outward part ($A \leftarrow B$) and backward part ($B \leftarrow A$), and stage changing point (A) as the TP. Second, on the account of approximate symmetry property, the velocity of the stage TP is zero which can be reflected evidently in the gyroscope channels. In order to recognize an activity, the Half-DTW in iCoach requires two steps. First, the Half-DTW applies an outward-backward segmentation operation to divide a repetition into outward part and backward part through the localization of TP. Second, based on the DTW algorithm, the Half-DTW technique applies the corresponding matching operation and provides a comprehensive output as the identified activity type.

Outward-Backward Segmentation: According to the property of approximate symmetry, this can easily be done by splitting a segment into two equal parts. However, in reality, the split segment cannot be ideally symmetric, and therefore causes a larger matching error according to the extension principle of DTW in the following matching phase. As a consequence, we propose to segment the outward and backward parts by finding the TP in gyroscope readings. As a result, the SP, TP, and EP in a repetition can be pointed in a data segment, which can further be utilized in detecting nonstandard behaviors and quality assessment. A sample illustration of the relationship among {SP, TP, EP}, outward-backward parts, and the arm movements in a repetition are shown in Fig. 5. The outward-backward segmentation is also implemented by a sliding window of size p (the same with the step length in activity segmentation) for the purpose of segmenting a whole activity, outward and backward parts simultaneously.

The iCoach has three steps for outward-backward segmentation. First, we perform data selection. Specifically, we only pick single-dimensional data (from gyroscope) dynamically among all the measurements for the localization of TP. The key insight is that gyroscope is superior in reflecting actual moving state (more precisely, rotary movement) compared to the accelerometer and the gyroscope velocity at the TP is zero.

However, different training activities weigh differently in the three axes. Intuitively, it is better to choose a dimension that possesses the most noticeable measurements so that it is much easier to detect the TP. As for this, we compare the fluctuation range of the 3-axis gyroscope and select the data from the axis which possesses the maximum fluctuation range for the following localization. Second, we detect the maximum peak velocity on the selected 1-axis and then TP must be situated between the Outward maximum peak velocity and Backward maximum peak velocity. Third, the minimum energy window between the maximum peak velocity is identified as the TP window because the samples near TP have very low speed. The energy can be represented by root mean square (RMS). Let f denote a window consisting of n samples and let s_i denote the i th sample value of the selected 1-axis signal, then the RMS of the window f is

$$\text{rms}(f) = \sqrt{\frac{\sum_{i=1}^n s_i^2}{n}}. \quad (5)$$

Once the TP in a segment is found, all the measurements can be split into outward and backward parts. By the aforementioned method, the TP corresponding to the movements can be detected as shown in Fig. 4. As we can see, the measurements' profiles in accordance with the kinematic interpretation and our method can effectively detect the TPs of each activity.

Activity Recognition: As mentioned previously, we match the data segments with the corresponding templates of each activity and choose the one that outputs the minimum DTW value as the recognized result. According to the principle of the DTW algorithm, DTW costs $O(mn)$ to compare two time series of lengths m and n . However, with outward-backward segmentation, we can apply the Half-DTW matching technique in activity recognition and reduce the practical running time cost to be half of that with the DTW, even though the theoretical time complexity remains $O(mn)$. The whole process of recognizing activities with Half-DTW is shown in Algorithm 2.

D. Analysis and Assessment

In order to recognize training activities, only coarse-level analysis is required. However, we step further to conduct more fine-grained analysis which mainly includes the detection of the nonstandard behaviors and the estimation of their quality.

1) Detection of Nonstandard Behaviors: In order to detect the nonstandard behaviors as listed in Section III, we first enlarge a segment (that represents a repetition) which has been divided into two parts. To detect cases 1–3, we make statistics of several parameters for each activity segment based on standard templates. These parameters include overall duration of a repetition ($T(\text{SP}, \text{EP})$), first half duration ($T(\text{SP}, \text{TP})$), and second half duration ($T(\text{TP}, \text{EP})$), standard deviations in windows that cover (SP→TP) ($\text{std}(\text{SP} \rightarrow \text{TP})$), (TP→EP) ($\text{std}(\text{TP} \rightarrow \text{EP})$), and (SP→EP) ($\text{std}(\text{SP} \rightarrow \text{EP})$). The iCoach pre-stores $T(\text{SP}, \text{EP})$ of each kind of fitness motion performed by coach, when $t(\text{SP}, \text{EP})$ of a repetition performed by a volunteer exceeds the range $[\kappa_1 \times T(\text{SP}, \text{EP}), \kappa_2 \times T(\text{SP}, \text{EP})]$, case 1 is detected. $T(\text{SP}, \text{EP})$ depends on exercise type and

Algorithm 2: Activity Recognition Algorithm**Input:** Motion templates:
$$\{T_{k,total}^i = \langle T_{k,l}^i, T_{k,r}^i \rangle, i=1, \dots, M, k=1, \dots, K,$$

$$T_{k,l}^i = \langle S_{k,l,1}^i, S_{k,l,2}^i, \dots, S_{k,l,D}^i \rangle,$$

$$T_{k,r}^i = \langle S_{k,r,1}^i, S_{k,r,2}^i, \dots, S_{k,r,D}^i \rangle\};$$

//K, M and D represents the number of exercise type, templates set and data dimension, respectively. Unkonwn Motion:

$$\{S_{L,total} = \langle S_{L,total,1}, S_{L,total,2}, \dots, S_{L,total,D} \rangle, \text{length}=N\};$$
Output: Identified motion index: $L \in \{1, \dots, K\}$;

```

1 Choose  $i$  with  $\max(\max(S_{L,total,i}) - \min(S_{L,total,i}))$  as AxisI
  for each  $i$  in  $\{gx, gy, gz\}$ ;
2 for  $i=1: \text{lapstep}: N$  do
3    $\text{row} = S_{L,total, \text{AxisI}}(i : i + \text{lapstep} - 1)$ ;
4    $\text{AvgV} = \text{mean}(\text{row})$ ;  $\text{RmsV} = \text{rms}(\text{row})$ ;
5 end
6 Select index corresponding to maximum and minimal
   $\text{AvgV}$ , the smaller index as MinI, the bigger index as
  MaxI;
7 Select index corresponding to minimal in
   $\text{RmsV}(\text{MinI}:\text{MaxI})$  as TranIndex;
8  $\text{TP} = (\text{TranIndex} + \text{MinI} - 1) * \text{lapstep}$ ;
   $S_{L,l} = S_{L,total}(1 : \text{TP})$ ;  $S_{L,r} = S_{L,total}(\text{TP} + 1 : N)$ ;
9 for each motion  $j=1$  to  $K$  do
10    $\text{TotalD}(j)=0$ ;
11   for each motion set  $i=1$  to  $M$  do
12     for each axis  $c$  in  $\{\text{accelerometer}, \text{gyroscope}\}$  do
13        $\text{TotalD}(j) = \text{DTW}(S_{L,l,c}, S_{j,l,c}^i) + \text{DTW}(S_{L,r,c}, S_{j,r,c}^i)$ ;
14     end
15   end
16 end
17 Select index corresponding to minimal TotalD as  $L$ ;
```

it has been preset in our system according to the standard statistics of coaches. Similarly, when $\max(t(\text{SP}, \text{TP}), t(\text{TP}, \text{EP})) / \min(t(\text{SP}, \text{TP}), t(\text{TP}, \text{EP}))$ exceeds the corresponding range, case 2 is asserted to be detected. When $\text{std}(\text{SP} \rightarrow \text{TP})$, $\text{std}(\text{TP} \rightarrow \text{EP})$ or $\text{std}(\text{SP} \rightarrow \text{EP})$ exceed the corresponding range, case 3 is asserted to be detected. iCoach reports the results to the user immediately after each repetition is finished for real-time guidance.

2) *Assessment of the Exercise Quality*: Due to the fact that exercise is usually taken in groups where each group contains a set of repetitive activities (group repetitions) in the strength training, we take both the single repetition and group repetitions into account for a more comprehensive assessment.

a) *Assessment of single repetition*: The assessment of single repetition aims to evaluate how well each repetition is performed and provides a more intuitive form to understand the performance of their exercise of that repetition. We utilize two metrics named *standard degree* and *execution duration* to monitor the quality of a repetition in real time. These two metrics are defined as follows.

1) *Standard Degree (SD)*: For the quantitative quality assessment of training exercise performed by volunteers,

we adopt the DTW value as a metric to evaluate the overall quality of a certain repetition. Intuitively, the DTW value indicates the similarity measure between an activity and the corresponding templates. When an activity performed by a volunteer is more similar to the corresponding templates, it shows that the quality of this performed activity is higher. We further propose SD to quantify the DTW value in the $[0, 10]$ range for providing users a more explicit response. The more standard the activity, the higher the SD. SD is defined as

$$\text{Standard Degree} = \left(1 - \frac{\text{DV}}{\text{DV}^*}\right) \times 10 \quad (6)$$

where DV is the calculated DTW value of a repetition. DV indicates the similarity between the performed repetition of a given exercise and its corresponding template. DV^* is the average DTW value calculated with the remaining templates in the same recognition process. The remaining templates can be regarded as completely mismatched, and so DV^* can be used as the current maximum dissimilarity measure. In iCoach, the Half-DTW value computed in the recognition process is a substitute for the DTW value.

2) *Execution Duration (ED)*: It measures how long a repetition is performed by the body builder. As mentioned above in Section III, nonstandard behaviors such as too-long or too-short ED of an exercise repetition should be avoided for building muscles effectively and preventing injuries. In addition, the closer ED to the standard duration indicates more standard activity. For an exercise type i , let T_i be the ED and T_s be the corresponding standard duration. T_i can be easily obtained from the segmentation result and T_s can be obtained from the coaches or the fitness guide book. Then, iCoach computes the difference $T_i - T_s$ and report to bodybuilders for speed adjustment. The more similar $T_i - T_s$ in a set of repetitions also indicates more regular movements and controls.

b) *Assessment of group repetitions*: The assessment of group repetitions aims to evaluate how well each group of repetitions are performed. Consistency plays a significant role in strength training [36]. We utilize two metrics, namely, *smoothness* and *continuity* as used in [10]. These two metrics are defined as follows.

1) *Smoothness*: Smoothness describes the consistency between repetitions in an activity group. More specifically, strength training strongly requires the exercisers to maintain accurate posture and control muscle carefully. The indicator can depict the similarity measure among the activities in a group. The greater the similarity measure is, the better the exercisers perform due to favorable muscle control. In the discrete signal series, let the probability distribution function of a repetition Rep1 is $\text{PDF}(\text{Rep1}) = \{p_i\}_{i=1}^m$, where p_i represents the probability value of the i th bin. In our measurements, we find that the value of SMA fluctuates in the range between 80 and 115. Therefore, for simplicity, the number of bins m is empirically set to 8 in order to obtain practical strength training experience. In order to attain the



Fig. 6. User interface of iCoach. (a) Moments module. (b) and (c) Training monitoring module. (d)–(f) Training assessment module.

similarity between activities, we adopt a metric, namely, Earth mover's distance (EMD) [37]. The EMD measures the minimal cost required to transform from one PDF to another one. As smaller EMD indicates a more similar data distribution of repetitions, therefore a better consistency of repetitions that the user performs.

- 2) *Continuity*: It describes the consistency between the rest intervals of an activity group. For more effective training, the bodybuilders have better retain the pace in an activity group. A more regular stretching and warping pace indicates a more powerful muscle control. The best scenario is that the interval of two motions stays the same within a group of activities. To quantify the consistency between the rest intervals of an activity group, we consider that the rest interval is a standard normal random variable. In order to assess the continuity of an activity group, we adopt the kurtosis as a metric. In statistics, the coefficient of kurtosis measures the sharpness of the peak in a variable distribution. More specifically, let $R = \{r_i\}_{i=1}^n$ be the vector of the rest intervals within an activity group. The kurtosis can be calculated as follows:

$$\beta = \frac{\sum_{i=1}^n (r_i - \mu)^4}{(\sum_{i=1}^n (r_i - \mu)^2)^2} - 3 = \frac{\mu^4}{\theta^4} - 3 \quad (7)$$

where μ and θ are the mean value and standard deviation of the rest interval vector. A larger kurtosis value indicates a more concentrated distribution of the data which indicates a better consistency of the rest intervals for a user. In iCoach, the obtained acceleration data are dependent on posture due to the effect of gravity. Consequently, we only compute the kurtosis value of 3-axis gyroscope data to evaluate the training performance of the user.

V. IMPLEMENTATION AND EXPERIMENTS

A. System Implementation

In this section, we describe both the hardware and software of iCoach.

Hardware: We implement a prototype of iCoach using commercial hardware. As for the IMU, we adopt a widely

used type, namely, MPU9250, a 9-DOF module that can output 3-axis accelerations, rotation speeds, and magnetometer. This unit is connected to a microcontroller for data collection and processing. The microcontroller is a low-cost board that supports connections of various sensors and enables C programming to control data collection and processing. Besides IMU, the microController is also connected to a Bluetooth 4.0 module for the purpose of data transmission to a mobile device.

Software: The APP is implemented on a Huawei P8 with an Android 6.0 operating system responsible for the main data-processing algorithms, as the computing capacity of a smartphone is more powerful than that of a microcontroller. The whole application is programmed with both Java and C codes. The Java codes are responsible for high-level logic and user interface design, while the C codes are utilized for speeding up low-level signal processing. The detection of non-standard behaviors and quality assessment results are displayed on the user interface. These results can also be reported to the users in the form of a voice reminder.

User Interface: The iCoach user interface currently provides three major modules for an exercise trainer, including moments module, training monitoring module, and training assessment module. Fig. 6 shows the screenshots of iCoach's user interface. The moments' module provides the mechanism to share training and daily moments with friends. The training monitoring module aims to provide the user with real-time guidance which includes basic statistics and assessment of individual repetition and repetition groups. Raw training statistics hold the raw data stream of exercise repetitions for the purpose of visualization. Moreover, the usage mechanism of this module is simple. Before strength training, the user only needs to put on the smart glove and press the start button to connect with the smart glove. Upon startup, this module starts the monitoring process and the user can put his cell phone aside while exercising. The training assessment module offers the quantitative daily training summary, including training intensity estimation, and assessment results of single and group repetitions of the ongoing activities. Intensity estimation is the completion indication of the training goal. The users can check their daily training intensity and assessment results anytime they want.

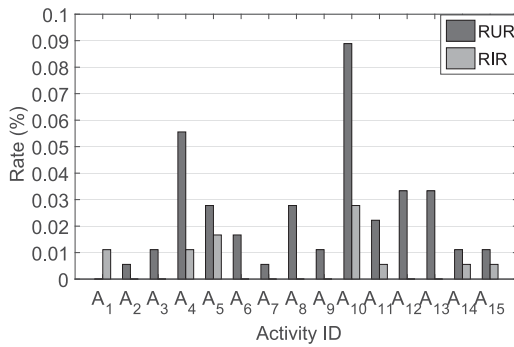


Fig. 7. Average RUR and RIR of different activities of iCoach.

B. Experiments Setup

We conduct experiments in a university gym to evaluate iCoach, and the corresponding experiment scenario is shown in Fig. 1. We recruit a total number of eight volunteers from our university who are with different levels of fitness knowledge and experience. Among them, three volunteers go to the gym regularly and the rest rarely go to the gym. As for the participants who have no idea about fitness activities, we let them turn to the fitness guide before experiments in order to make sure that they can totally understand how experiments should be conducted. We make a schedule of the 15 training activities and ask the volunteers to perform each activity for 20 sets, with ten repetitions in each set. As a result, we can obtain a total of 24 000 [8 (participants) \times 15 (activities) \times 20 (sets) \times 10 (repetitions)] test repetitions. Each sensor collects measurements with a sampling rate of 40 Hz. Besides, we recruit one physically strong coach with years of training experience to follow the same schedule under the same settings in order to provide standard templates. During the course of experiments, we take videos of entire processes as validating materials in the evaluation part. As the evaluation baseline, we also collect the volunteers' own standard templates under the guide of the coach.

VI. PERFORMANCE EVALUATION

We mainly evaluate the performance of iCoach from five different aspects, including segmenting activities, recognizing activities, detecting nonstandard behaviors, assessing exercise quality, and user-study experience.

A. Evaluation on Activity Segmentation

We evaluate the performance of detecting activities with two main metrics of repetition undetection rate (RUR) and repetition insertion rate (RIR). The definition of these two metrics is explained in detail as follows.

- 1) *RUR*: If an activity is executed for N times and it is correctly detected for n times, the Rep Undetection Rate is defined as $(N - n/N)$.
- 2) *Repetition Detection Rate (RDR)*: It is equal to $1 - \text{RUR}$.
- 3) *RIR*: If an activity is executed for N times and iCoach detects that within the rest interval for n times, the Rep Insertion Rate is defined as (n/N) .

Fig. 7 shows RUR and RIR across 15 tested programs. As we can see, the average RUR is 2.41% while the average RIR is

0.55%. In order to evaluate iCoach in a comprehensive manner, we show that the average RDR reaches 97.59% according to the average RUR. The highest RUR occurs at A_{10} , but RDR still remains around 91% as shown in Fig. 7. This is mainly caused by the relatively small range of motion movement. The results indicate that iCoach can detect repetitions accurately, and such high RDR makes subsequent fine-grained statistical feedback possible and reliable.

B. Evaluation on Activity Recognition

For an automated gym system, the ability to identify activities with higher recognition accuracy and higher time efficiency is pretty important. We utilize the recognition accuracy and time delay as the major evaluation metrics. These two metrics are explained in detail as follows.

- 1) *Accuracy*: If an activity is executed for N times and it is correctly recognized for n times, the recognition accuracy is defined as (n/N) .

- 2) *Computational Delay*: The duration of time required to sense an activity while it is being executed.

1) *Comparison of Activity Recognition With DTW, Fast-DTW, and Half-DTW*: We first evaluate the performance of three recognition algorithms based on DTW, Fast-DTW, and Half-DTW techniques. Fig. 8(a) shows the average recognition accuracy of each exercise type using three recognition algorithms and the same set of templates from the coach. Similarly, Fig. 8(b) shows the average recognition time delay. Specifically, the average accuracy of the aforementioned 15 exercise types based on the DTW, Fast-DTW, and Half-DTW techniques are 97.37%, 95.18%, and 96.07%, respectively. Moreover, the average time delays using different algorithms are 174.66, 185.30, and 103.76 ms, respectively. As we can see, the DTW technique achieves the highest accuracy and the Fast-DTW technique incurs the lowest one. This is because the Fast-DTW algorithm has a certain extent of precision loss because of the search space optimization. Therefore, the Fast-DTW technique suffers a significant decline in accuracy. Nevertheless, we also observe that the Fast-DTW does not achieve the time optimization as we expected in general. This is because the time series of strength training activities are usually short, which makes the cost of projecting and refining a warping path recursively exceed the cost of calculating DTW directly. Our experiments prove that the time cost of the Fast-DTW is slightly better compared to the DTW when the exercise motion is very slow. As a result, obviously, the Fast-DTW is not suitable for our problem. We also find that the average accuracy achieved by the Half-DTW is slightly lower compared with that by the DTW, but is still up to 96.07%. However, in terms of time efficiency, we can notice that the Half-DTW incurs remarkably smaller time delay in all situations which is nearly half of that by DTW. Taking both accuracy and time efficiency into consideration, we finally employ the Half-DTW as our recognition algorithm. Overall, these results demonstrate that iCoach can recognize 15 types of activities with high accuracy and low time delay. What is more, our system achieves better recognition accuracy and time efficiency performance comparing with several similar existing

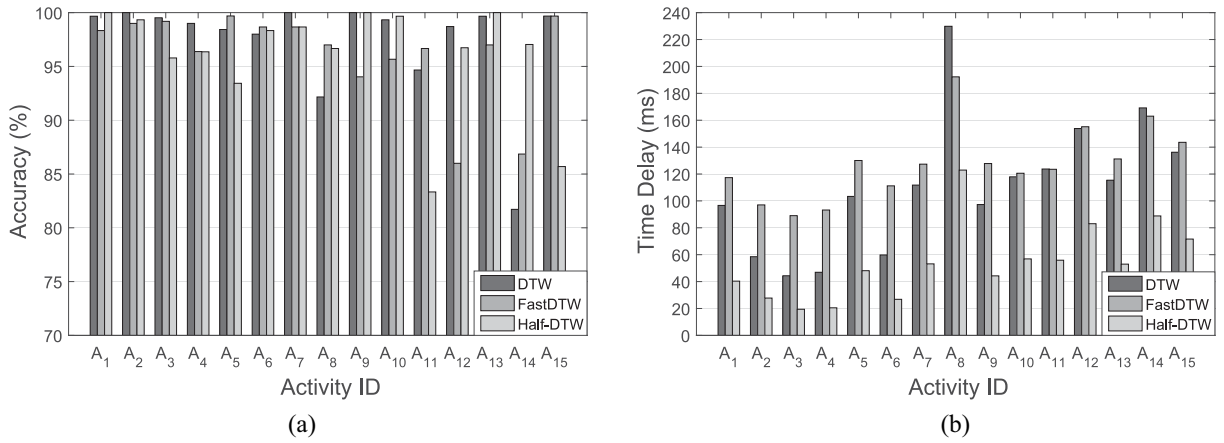


Fig. 8. Recognizing performance comparison among the DTW, Fast-DTW, and Half-DTW techniques. (a) Recognition accuracy comparison. (b) Recognition time delay comparison.

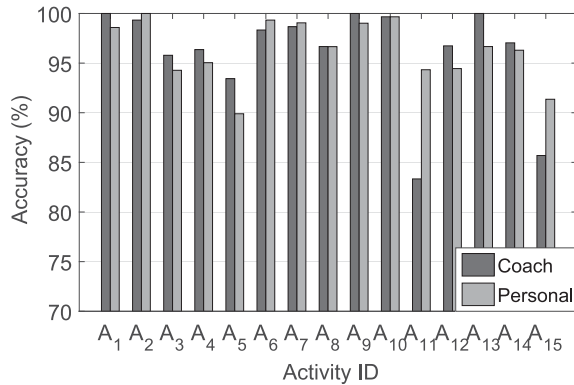


Fig. 9. Results of activity recognition when matching templates are from a coach and users themselves.

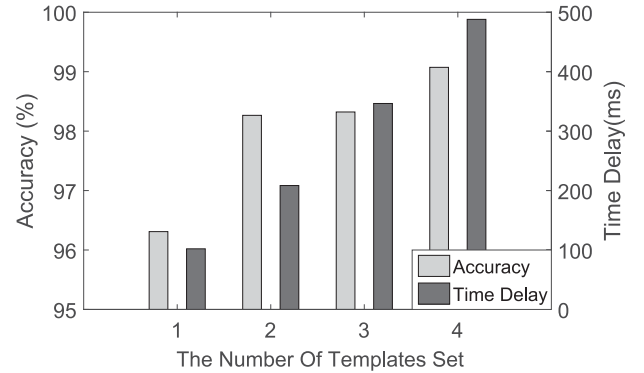


Fig. 10. Impact of the number of templates on recognition accuracy and computational delay.

systems, such as 90% accuracy achieved by FEMO [10] for ten fitness types, 87% accuracy by Xie *et al.* [14] for ten fitness types, and 95% accuracy by Guo *et al.* [13] for 12 fitness types.

2) *Sensitivity to the Source of Templates:* Next, we evaluate the influence of template sources on the accuracy. Fig. 9 shows the corresponding outcome when the prestored templates come from the coach or exerciser themselves using only one set of templates. We can see that the average accuracy of the aforementioned 15 exercise types using only one set of templates from the coach and the exerciser themselves are 96.07% and 96.31%, respectively. Despite lower accuracy compared to the other one, in reality, the usage of coach templates is acceptable as the accuracy is still up to 96.07%. Moreover, for the bodybuilders who achieve different stages of fitness certificate (i.e., novice exercisers and experienced exercisers), it is required to replace the templates individually in order to obtain higher accuracy and reasonable fitness assessment. Therefore, the usage of templates from a coach is more suitable for effective fitness assessment in our system.

3) *Sensitivity to the Number of Template Sets:* We further investigate the impact of the number of template sets. We evaluate the overall recognition accuracy and average time delay

of recognizing 15 exercise when the number of template sets varies from 1 to 4 as shown in Fig. 10. As the number of templates increases, the accuracy increases and reaches a stable value of 98.3% when two or more templates are utilized. Nevertheless, the average time delay also grows linearly with the number of templates, from less than 100 ms to almost 500 ms. We notice the fact that even when one set of template is used, the recognition accuracy still reaches 96.1%, while the time delay is only about 90 ms, nearly half of that with two sets. The real-time performance is almost the same with FEMO [10] which is implemented on a Lenovo PC. Considering the tradeoff between time delay and recognition accuracy, we only make use of one set of templates in our system.

4) *Matching Rate Among Multiple Activities:* Finally, we investigate the confusion matrix achieved by the Half-DTW technique using only one set of coach templates in Fig. 11. We observe that there are mismatches in some activities, such as A5 and A12, A1 and A15. This may be caused by the similarity movement trails of the corresponding two activities themselves. We can consider the utilization of a hierarchical recognition framework in the future for further performance enhancement. Fortunately, with the current setup, we can still achieve a higher recognition accuracy overall.

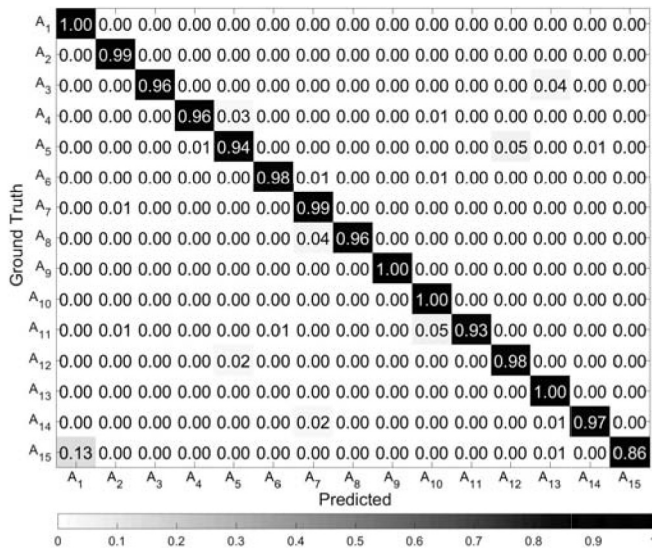


Fig. 11. Confusion matrix achieved by the Half-DTW technique with one set of coach templates.

TABLE I
PERFORMANCE IN DETECTING NONSTANDARD BEHAVIORS

Metric \ BehaviorID	1	2	3
Precision (%)	92.5	89.0	90.5
Recall (%)	89.0	91.5	90.5

C. Detection of Nonstandard Behaviors

It is possible that the abnormality detection system can miss, confuse, or falsely detect activities. Consequently, two metrics, namely, precision and recall are employed to evaluate the performance of detecting abnormal behaviors. The definition of two metrics is explained in detail as follows.

- 1) *Precision*: In activity recognition, *precision* is defined as $(TP/TP + FP)$, in which TP and FP represent true positive and false positive, respectively.
- 2) *Recall*: It is defined as $(TP/TP + FN)$, in which TP and FN represent true positive and false negative, respectively.

However, as it is difficult to label nonstandard behaviors during the course of exercise, we request the coaches to carefully watch recorded videos to aid in obtaining ground truth. Table I presents the performance of detecting each nonstandard behavior. The average precision of detecting these behaviors is 90.7% and the corresponding recall rate is up to 90.3%. In a nutshell, iCoach can detect the aforementioned three types of nonstandard behaviors during the course of strength training with acceptable accuracy.

D. Assessment of Exercise Quality

For evaluating the reliability of assessing process, we have conducted comprehensive experiments under three main aspects which are described as follows.

1) *Validity Evaluation of Standard Degree Metric*: Experienced volunteers were asked to perform each exercise type 20 times in two different forms: strictly following

the standard fitness specification (R_s for short), with obvious interference factors such as shake but trajectory profiles of activities still coincide with the standard fitness specification (R_n for short). On account of the shake of arms, the muscles cannot maintain certain required tension. Therefore, we consider the second exercise form as nonstandard. We present the corresponding results in Fig. 12(a) in terms of this metric. As we can see, the Standard Degree metric of nonstandard activities is obviously lower compared to the standard activities in general, but there are still some similar SD for some exercise type, such as A_4 and A_{13} . This may be caused by the fact that these exercise types are performed with corbel of gym equipment and it is hard for volunteers to shake. Overall, the Standard Degree metric in iCoach can reflect the fitness quality of a repetition effectively.

2) *Validity Evaluation of Group Repetitions Assessment Metrics*: We evaluate the rationality and validity of group repetitions assessment metrics, namely, *smoothness* and *continuity*. In this section, expert volunteers were asked to perform activities in the aforementioned two forms (R_s and R_n). For the sake of convenience, we define ten repetitions as a group and volunteers were asked to perform two groups one after another. Ten repetitions in the former group (i.e., group1) were performed in the $\{R_s, R_s, R_s, R_s, R_s, R_s, R_s, R_s, R_s, R_s\}$ form. Meanwhile, volunteers were asked to remain stable during the rest intervals. The latter group (namely, group2) were performed in the $\{R_s, R_s, R_s, R_s, R_n, R_s, R_n, R_n, R_n, R_n\}$ form. At the same time, the volunteers were asked to remain stable and shake in turns during the rest intervals. The volunteers were asked to perform each group 20 times. As an outcome of these experiments, the corresponding *smoothness* results of the groups are shown in Fig. 13. In the result, the EMD values between the adjacent repetitions in group1 are all lower. As smaller EMD value indicates a similar probability distribution, it indicates a more regular arm movement for the people in group1. However, as for group2, that the EMD values are obviously larger in R_s and R_n forms alternatively indicates different probability distribution between R_s and R_n , and therefore it indicates a less regular arm movement in alternations of group2. Meanwhile, despite the EMD values of the consecutive R_s and R_n forms in the former and latter part, respectively, are also small, we can get a comprehensive group repetitions assessment by combining the *smoothness* metric with the Standard Degree metric. The corresponding *continuity* results are shown in Table II. The kurtosis of group1 is obviously larger than that of group 2. As larger kurtosis indicates a more concentrated distribution, it indicates more stable rest intervals of the people in group1. Thus, the *smoothness* and *continuity* results of the experiments coincide with the theoretical analysis.

3) *Feedback Comparison Between Real Coach and iCoach*: Finally, we compare the feedback provided by the real coaches with that by iCoach. We record the average value of *Standard Degree* of each participant across all activities. Meanwhile, we request the coaches to watch videos of each volunteer and grade the quality of each set of activities in the [1, 10] range. When all sets of activities for all volunteers are graded, we average each volunteer's grades as his final score. We compare

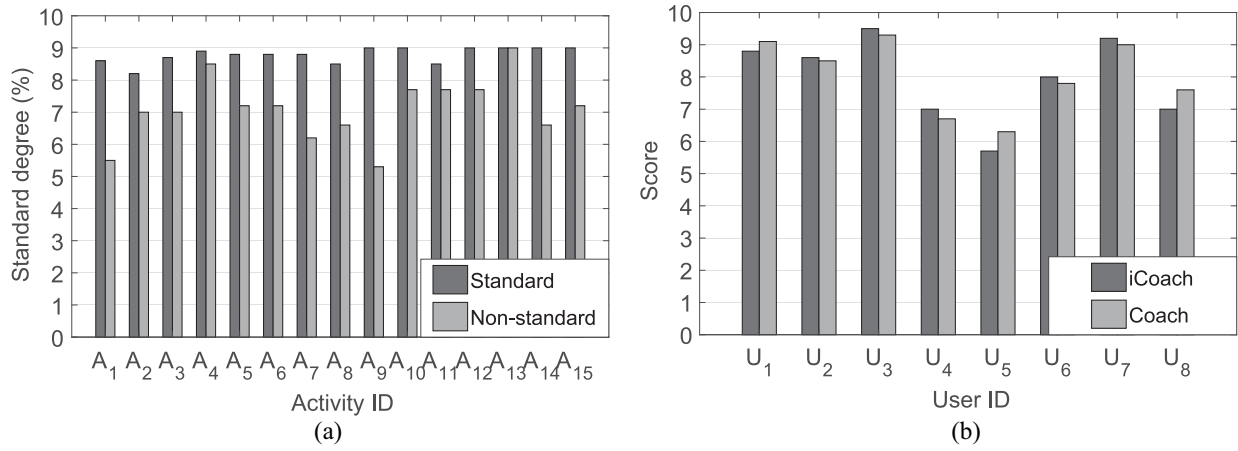


Fig. 12. Sample illustration of assessing exercise quality. (a) Comparison of SD given by iCoach between the standard and nonstandard forms. (b) Score comparison between coaches and iCoach.

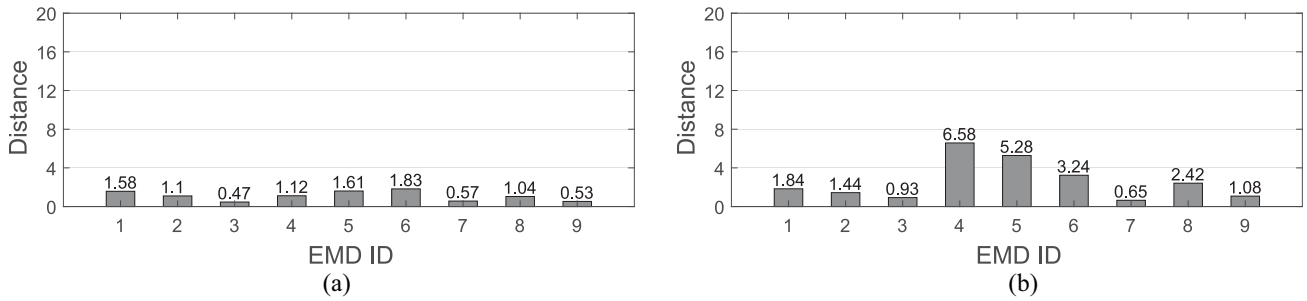


Fig. 13. Sample illustration of smoothness result. (a) Group1. (b) Group2.

TABLE II
SAMPLE CONTINUITY ASSESSMENT RESULTS OF GROUP1 AND GROUP2

Group ID	Metric ID		
	Gx	Gy	Gz
Group1	-2.92	-3.00	2.99
Group2	-3.00	-3.00	-2.09

the average score of each volunteer provided by iCoach and that by the real coach in Fig. 12(b). Although the specific quantitative scores are different, the ranks provided by iCoach and the real coach is rather similar. This indicates the feasibility and reliability of assessing training exercise using iCoach.

E. Run-Time Performance

1) *Evaluation on the Feedback Provided by iCoach:* In order to evaluate the quality of user experience, we conduct a survey on eight volunteers in order to obtain feedback on the practicability and usage capability based on their real-time experience. We ask the participants to answer four questions. For each question, a volunteer is required to provide a score between 1 (lowest) and 10 (highest). Once the survey is done, we utilize these scores to reflect the satisfaction of the volunteers.

Table III shows the overall feedback outcome on iCoach. The volunteers suggest that iCoach could accelerate the training objective with an average score of 8.57 and 0.97 standard deviation. Comparing with traditional coaches, iCoach

TABLE III
FEEDBACK ON iCOACH

Feedback item	Rating	Std
Can iCoach help to train efficiently	8.57	0.97
Compared to the real coaches, do iCoach have advantages	8	1.53
Would the training be continued because of the quantitative data	9.14	1.21
The detected non-standard activities do help to train effectively	8.57	0.78

achieves an enhanced score of 8 on average with 1.53 standard deviation. Moreover, the participants are encouraged to continue training while receiving the quantitative feedback from iCoach. Last but not least, the participants also benefit significantly from the detected nonstandard activities with an average score of 8.57 and 0.78 standard deviation. Overall, the scores from volunteers verify that iCoach can assist in efficient training activities and maintain a regular training habit over the long term.

2) *CPU Occupation:* We also evaluate the occupancy of CPU resources occupation when the smartphone merely runs iCoach. In the evaluation process, we shut down all the other applications and monitor CPU consumption in the USB debugging mode. Meanwhile, we request a user wearing the glove to perform fitness exercise continuously to test the maximum CPU consumption required for iCoach. The whole testing process lasts for 5 min. Fig. 14 shows the proportion of CPU time consumed during the course of recognizing activities. Even

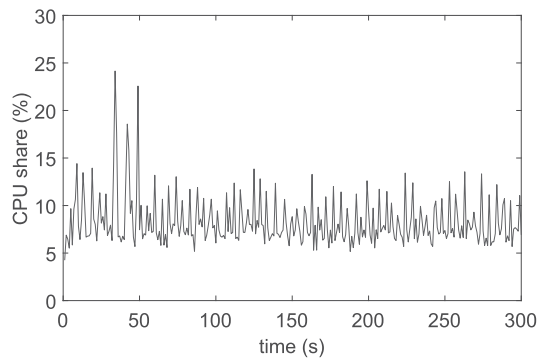


Fig. 14. CPU overhead during the course of recognizing activities.

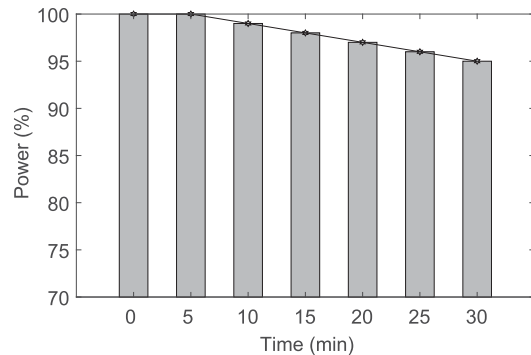


Fig. 15. Energy consumption when iCoach is in the running state.

though the CPU occupation percentage varies from 4.25% to 24.15%, the average value is only 8.26% with a standard deviation of 2.55% which is acceptable for an application.

3) *Energy Consumption*: We also conduct experiments to evaluate the energy consumption of iCoach application in the smartphone end. We fully charge the Huawei P8 smartphone, kill all the other programs, and run the iCoach application continuously. During testing periods, the screen is turned off after making necessary settings. Similar to experiments in the above, a user wearing the smart glove connected with the smartphone performs fitness exercises, such as seated dumbbell curl, hammer curl, and dumbbell fly for a period of 30 min. The real-time power level of the smartphone is obtained through the Android API and shown in Fig. 15. As we can see, the power level decreases from 100% to 95% after 30 min, which implies that about 0.84% power is consumed every 5 min. This power consumption has a negligible impact on the daily use of the smartphone.

VII. DISCUSSION AND FUTURE WORK

There are several points to be discussed as follows.

Question 1: Why Not Use Deep Learning Models? Deep learning (DL) models are increasingly used in various kinds of activity recognition and achieve satisfactory performance [38]. Thus, a straightforward idea to design such a system is to adopt DL models, such as CNN, RNN, or LSTM as done in previous works [39], [40]. However, we do not follow this approach based on the following considerations. First, DL models usually require a huge amount of training samples in order to avoid overfitting, which brings about an

intensive labor overhead of collecting the training data. This decreases the system's practicability in the real world. In contrast, our work follows a totally training-free approach. Once we collect several standard templates from a coach and store them on a smartphone, other users can directly use the system without training the system. It is noted that although we can train a general DL model with samples collected from certain persons and apply it to others, in which way the system is training free for them, it usually requires more complex techniques, such as transfer learning [41], [42], generative adversarial network [43], [44], and deep active learning [45], [46], and thus still needs collecting massive data for initial model training. What is worse, for the sake of high performance, this general model shall cause rather heavy computational overhead and energy burden consumption which is beyond the power of a mobile device. By utilizing several samples as templates from the same coach, our approach can achieve a high accuracy of 96% and less than 90-ms real-time performance. Considering the tradeoff between accuracy and resources requirement, we think that our approach is a more favorable choice.

Question 2: Can This System Be Extended to Recognize More Activities? In the current version of iCoach, we consider 15 kinds of strength training activities that only involve upper body parts. Although these activities cover most upper body exercises, there are more different categories of activities especially those involving the lower body, such as seated leg curl and lying leg curls. It is to be admitted that only utilizing the IMU sensors worn in the upper body, it is impossible to recognize those lower body strength training exercises. This is a limitation of our system. However, we envision that by equipping a user with the same sensors in shoes, the data processing techniques can be extended to recognize lower body training activities. The underlying reason is that similar to upper body activities, the lower body ones have similar cyclic patterns which can be reflected in the IMU measurements. What is more, with the increasing popularity of wearable devices such as smart shoes, we believe that this does not induce much additional hardware costs. We leave the extension of our system as one of the future work.

Question 3: Can This System Be Improved to Monitor More Fine-Grained Information of Activities? Someone may wonder why we design a smart glove system instead of utilizing other wrist-worn smart devices such as a smartwatch to achieve the same goals. Actually, there are two main considerations behind this choice. For one thing, using a smart glove system is more practical than a smartwatch. This is because a glove is a necessity for weight training exercise to prevent hand injuries. It is a more practical choice to add removable and low-cost hardware plug-ins to transform it to a smart device and enrich its functions, compared with utilizing a smartwatch. For another thing, based on the glove, a more comprehensive assessment scheme of doing weight training exercise can be further explored. For example, by integrating commercial pressure and physiological sensors such as heart rate meter, EMG electrode with the current version of iCoach, multimodal data of the human body during the course of weight training exercise can be obtained and utilized for more thorough analysis. By this means, besides

knowing about the type, quantity, and quality of doing exercise, richer information such as calorie consumption can be acquired. We admit that the current version of the smart glove is limited to monitor relatively coarse-level information of strength training exercise.

VIII. CONCLUSION

In this article, we propose a novel low-cost system, iCoach, for strength training enthusiasts, which can act as a professional fitness instructor and provide real-time monitoring and coaching service for them. The iCoach is built on a standard accessory, i.e., fitness glove, with pervasive inertial sensors embedded in an unobtrusive manner. Apart from recognizing types of activities, the iCoach zooms in each subprocess in a repetitive manner to detect nonstandard behaviors and provides timely feedback for the users if abnormal events are detected. Moreover, based on obtained motion data, the iCoach assesses qualities of users' exercise. Experiments in a university gym demonstrate that iCoach recognizes 15 categories of activities, detects nonstandard behaviors, and assesses exercising quality with favorable performance.

ACKNOWLEDGMENT

The authors really appreciate the kind effort of the Associate Editor and anonymous reviewers, for giving precious suggestions to improve this article.

REFERENCES

- [1] T. R. Baechle and R. W. Earle, *Essentials of Strength Training and Conditioning*. Champaign, IL, USA: Human Kinetics, 2008.
- [2] R. Frederick. (2015). *Multi-Year Weight Training*. Accessed: Feb. 26, 2020. [Online]. Available: <http://www.weightliftingapp.com/>
- [3] S. Rocca. (2015). *Gymbook*. Accessed: Feb. 26, 2020. [Online]. Available: <https://www.gymbookapp.com/>
- [4] S. B. Eisenman *et al.*, "BikeNet: A mobile sensing system for cyclist experience mapping," *ACM Trans. Sensor Netw.*, vol. 6, no. 1, p. 6, 2009.
- [5] N. Y. Hammerla, T. Plötz, P. Andras, and P. Olivier, "Assessing motor performance with PCA," in *Proc. Int. Workshop Front. Activity Recognit. Pervasive Sens.*, 2011, pp. 18–23.
- [6] Fitbit Inc. (2015). *Fitbit*. Accessed: Feb. 26, 2020. [Online]. Available: <https://www.fitbit.com/cn>
- [7] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, "Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach," in *Proc. IEEE ISWC*, 2012, pp. 17–24.
- [8] A. Zhan, M. Chang, Y. Chen, and A. Terzis, "Accurate caloric expenditure of bicyclists using cellphones," in *Proc. ACM SenSys*, 2012, pp. 71–84.
- [9] R. Chaudhri, J. Lester, and G. Borriello, "An RFID based system for monitoring free weight exercises," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 431–432.
- [10] H. Ding *et al.*, "FEMO: A platform for free-weight exercise monitoring with RFIDs," in *Proc. ACM SenSys*, 2015, pp. 141–154.
- [11] K.-H. Chang, M. Y. Chen, and J. Canny, "Tracking free-weight exercises," in *Proc. Int. Conf. Ubiquitous Comput.*, Springer, 2007, pp. 19–37.
- [12] D. Morris, T. S. Saponas, A. Guillory, and I. Kelner, "RecoFit: Using a wearable sensor to find, recognize, and count repetitive exercises," in *Proc. ACM SIGCHI Conf. Human Factors Comput. Syst.*, 2014, pp. 3225–3234.
- [13] X. Guo, J. Liu, and Y. Chen, "FitCoach: Virtual fitness coach empowered by wearable mobile devices," in *Proc. IEEE INFOCOM*, 2017, pp. 1–9.
- [14] L. Xie, X. Dong, W. Wang, and D. Huang, "Meta-activity recognition: A wearable approach for logic cognition-based activity sensing," in *Proc. IEEE INFOCOM*, 2017, pp. 1–9.
- [15] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis, "RISQ: Recognizing smoking gestures with inertial sensors on a wrist-band," in *Proc. ACM Mobisys*, 2014, pp. 149–161.
- [16] S. Shen, H. Wang, and R. R. Choudhury, "I am a smartwatch and I can track my user's arm," in *Proc. ACM Mobisys*, 2016, pp. 85–96.
- [17] E. Thomaz, I. A. Essa, and G. D. Abowd, "A practical approach for recognizing eating moments with wrist-mounted inertial sensing," in *Proc. ACM Ubicomp*, 2015, pp. 1029–1040.
- [18] T. Hao, G. Xing, and G. Zhou, "RunBuddy: A smartphone system for running rhythm monitoring," in *Proc. ACM Ubicomp*, 2015, pp. 133–144.
- [19] H. Guo, L. Chen, L. Peng, and G. Chen, "Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble," in *Proc. ACM Ubicomp*, 2016, pp. 1112–1123.
- [20] A. Khan *et al.*, "Beyond activity recognition: Skill assessment from accelerometer data," in *Proc. ACM Ubicomp*, 2015, pp. 1155–1166.
- [21] C. Karatas *et al.*, "Leveraging wearables for steering and driver tracking," in *Proc. IEEE Infocom*, 2016, pp. 1–9.
- [22] S. Jain, C. Borgiattino, Y. Ren, M. Gruteser, Y. Chen, and C. F. Chiasserini, "Lookup: Enabling pedestrian safety services via shoe sensing," in *Proc. ACM MobiSys*, 2015, pp. 257–271.
- [23] M. Rabbi, M. H. Aung, M. Zhang, and T. Choudhury, "MyBehavior: Automatic personalized health feedback from user behaviors and preferences using smartphones," in *Proc. ACM Ubicomp*, 2015, pp. 707–718.
- [24] J. Lester, C. Hartung, L. Pina, R. Libby, G. Borriello, and G. Duncan, "Validated caloric expenditure estimation using a single body-worn sensor," in *Proc. ACM Ubicomp*, 2009, pp. 225–234.
- [25] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures," in *Proc. ACM Mobicom*, 2014, pp. 617–628.
- [26] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proc. ACM Mobicom*, 2013, pp. 27–38.
- [27] X. Zheng, J. Wang, L. Shanguan, Z. Zhou, and Y. Liu, "Smokey: Ubiquitous smoking detection with commercial WiFi infrastructures," in *Proc. IEEE Infocom*, 2016, pp. 1–9.
- [28] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proc. ACM Mobicom*, 2015, pp. 65–76.
- [29] X. Guo, J. Liu, C. Shi, H. Liu, Y. Chen, and M. C. Chuah, "Device-free personalized fitness assistant using WiFi," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 4, p. 165, 2018.
- [30] M. H. Stone, M. Stone, and W. A. Sands, *Principles and Practice of Resistance Training*. Champaign, IL, USA: Human Kinetics, 2007.
- [31] Weider Publications. *Muscle & Fitness*. Accessed: Feb. 26, 2020. [Online]. Available: <https://www.muscleandfitness.com>
- [32] Y. Verkhoshansky, *Fundamentals of Special Strength-Training in Sport*. Livonia, MI, USA: Sportivny Press, 1986.
- [33] W. J. Kraemer and N. A. Ratamess, "Fundamentals of resistance training: Progression and exercise prescription," *Med. Sci. Sports Exercise*, vol. 36, no. 4, pp. 674–688, 2004.
- [34] P. Aggarwal, Z. Syed, X. Niu, and N. El-Sheimy, "A standard testing and calibration procedure for low cost MEMS inertial sensors and units," *J. Navig.*, vol. 61, no. 2, pp. 323–336, 2008.
- [35] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.
- [36] C. Kennedy-Armbruster and M. Yoke, *Methods of Group Exercise Instruction*. Champaign, IL, USA: Human Kinetics, 2014.
- [37] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.
- [38] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges and opportunities," 2020. [Online]. Available: [arXiv:2001.07416](https://arxiv.org/abs/2001.07416).
- [39] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–28, 2017.
- [40] V. Radu *et al.*, "Multimodal deep learning for activity and context recognition," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–27, 2018.
- [41] S. A. Rokni, M. Nourollahi, and H. Ghasemzadeh, "Personalized human activity recognition using convolutional neural networks," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 8143–8144.
- [42] K. Chen, L. Yao, D. Zhang, X. Chang, G. Long, and S. Wang, "Distributionally robust semi-supervised learning for people-centric sensing," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3321–3328.

- [43] E. Soleimani and E. Nazerfard, "Cross-subject transfer learning in human activity recognition systems using generative adversarial networks," 2019. [Online]. Available: arXiv:1903.12489.
- [44] J. Wang, Y. Chen, Y. Gu, Y. Xiao, and H. Pan, "SensoryGANs: An effective generative adversarial framework for sensor-based human activity recognition," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, 2018, pp. 1–8.
- [45] H. S. Hossain, M. A. A. H. Khan, and N. Roy, "DeActive: Scaling activity recognition with active deep learning," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 2, pp. 1–23, 2018.
- [46] H. S. Hossain and N. Roy, "Active deep learning for activity recognition with context aware annotator selection," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2019, pp. 1862–1870.



Yongpan Zou (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2017.

Since September 2017, he has been working as an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His current research interests mainly include wearable/mobile/ubiquitous computing and HCI. For more information, see <https://yongpanzou.github.io/>.



Dan Wang is currently pursuing the master's degree with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

Her research interest covers mobile and ubiquitous computing.



Shicong Hong is currently pursuing the postgraduation degree with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

His research interest is mobile computing.



Rukhsana Ruby (Member, IEEE) received the master's degree from the University of Victoria, Victoria, BC, Canada, in 2009, and the Ph.D. degree from the University of British Columbia, Vancouver, BC, Canada, in 2015.

She has authored nearly 50 technical papers of well-recognized journals and conferences. Her research interests include the management and optimization of next-generation wireless networks.



Dian Zhang (Member, IEEE) received the Ph.D. degree in computer science and engineering from Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2010.

She worked as a Research Assistant Professor with the Fok Ying Tung Graduate School, HKUST. From 2012 to 2019, she worked as an Associate Professor with Shenzhen university, Shenzhen, China. She is currently an Associate Professor with Lingnan University, Hong Kong. Her research interests include big data analytics and mobile computing.



Kaishun Wu (Member, IEEE) received the Ph.D. degree in computer science and engineering from Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2011.

He worked as a Research Assistant Professor with HKUST. In 2013, he joined Shenzhen University as a Distinguished Professor. He has coauthored two books and published over 100 high-quality research papers in international leading journals and primer conferences, such as the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS

ON PARALLEL AND DISTRIBUTED SYSTEMS, ACM MobiCom, and IEEE INFOCOM. He is the inventor of six U.S. and over 100 Chinese pending patents.

Dr. Wu received the Hong Kong Young Scientist Award in 2012, the Hong Kong ICT Awards: Best Innovation in 2014, and the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2014.