Doctoral Dissertation

Ubiquitous Systems for Counting Intake and Burn out Calories.
摂取・消費カロリー推定のためのユビキタスシステム

Akpa Akpro Elder Hippocrate

March 15, 2019

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Akpa Akpro Elder Hippocrate

Thesis Committee:

Professor Keiichi Yasumoto	(Supervisor)
Professor Yoshinobu Sato	(Co-supervisor)
Associate Professor Yutaka Arakawa	(Co-supervisor)
Assistant Professor Hirohiko Suwa	(Co-supervisor)

Ubiquitous Systems for Counting Intake and Burn out Calories.
摂取・消費カロリー推定のためのユビキタスシステム*

Akpa Akpro Elder Hippocrate

Abstract

According to the World Health Organization latest statistics, nearly 39% of the world adults aged 18 years and over were overweight in 2016, and 13% were obese. In recent years, almost half of the global adults have attempted to control their weight in order to maintain a fit weight or to loss weight. To reach that goal, the most common methods are physical exercises and intake calorie restrictions. Weight loss occurs when the energy of food intake is less than energy expenditure. Therefore, to successfully manage body weight, it is necessarily to accurately estimate the amount of calorie expended through exercise and the calorie obtained from meals. In other words, for one's to control his/her body weight, it would be important to measure the intake food calorie and adjust it to the amount of expended calorie. However, previous studies have showed that in general people tend to underestimate calorie from food and overestimate calorie burned from physical activity.

The recent technology progress in the area of ubiquitous systems and wearable devices is perceived by the research community as an opportunity to monitor and measure food intake beyond clinical boundaries, as well as an opportunity to track and assess physical exercises. In this dissertation, we investigate two ubiquitous systems that can be used separately or together to assess calorie intake and calorie expenditure. The first system, which is an image-based system, estimates food weight and calorie from a single picture of the food by using ordinary

*Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, March 15, 2019.

eating-chopsticks as a measurement reference. This system requires the user to take only a single picture of his/her food, from the top with the chopsticks in the picture. Using several image processing techniques, and meta-data of the food image, the system automatically estimates the diameter and the height of the food container. Then, given the food type, the system combines the information about the container diameter, height and the food type to provide the weight of the food in the image, and finally estimates the calories of the food.
The second system recognizes and counts physical exercises via a wearable smart-glove. This system integrates 16 force sensitive resistor (FSR) sensors into wearable fitness gloves to assess physical exercises, by analyzing the time series of the pressure distribution in the hand palms of the user observed during workout sessions.
To assess the performances of the proposed systems, we ran two separated experiments of different types. For the food intake calorie estimation system, the experiment was ran over 15 food types and the system achieved an average relative error rate of 6.65% for the weight measurement and 6.70% relative error for the calorie estimation.
The results of the experimental trial on the smart-glove system conducted with 10 participants over 10 common fitness exercises showed 88.90% of F-score for overall activity recognition, in the case of user-dependent activity recognition. In the case of leave-one-participant-out cross-validation, the result showed 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.

Keywords:

Overweight and Obesity, Calorie estimation systems, Food intake, Physical activity, Image processing, Sensors, IoT.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

Overweight and obesity are described as an abnormal and/or excessive fat accumulation that may cause prejudice to one's health. According to the World Health Organization (WHO) recent updates published on February 2018 [1], the global rate of overweight and obesity have nearly triple since 1975. In 2016, 39% of world adults were overweight and 13% were obese. These increasing rates are a major public health concern. Both overweight and obesity are the fifth leading risk of global death, with at least 2.8 million adults dying each year as a result of being overweight or obese [2]. Since the fundamental cause of obesity and overweight is the imbalance between calories consumed and calories expended, the WHO preventions' methods against overweight and obesity recommend at the individual level the following 2 actions:

- Adopt a healthy eating behavior and control intake calories,
- Engage in regular physical activity.

Healthy eating involves eating calorie balanced meals, that is to say meals within a personal calories goal, which is calculated based on weight, height, and physical activity level. The recommended activity level is 60 minutes a day for children and for adults 150 minutes of moderate intensity aerobic exercises or 75 minutes of high-intensity cardio exercises every week. Thus, to successfully manage body weight and prevent obesity, it would be beneficial for an individual to be able to estimate: (1) the number of calories consumed in a meal and (2) the number of calories expended through physical exercises [3].
The clinical methods to prevent overweight and obesity require the patients to measure, record, and report daily food intake. Then, the daily recorded food calorie intake is then compared again an estimated calorie burned from physical activity, to provide sufficient feedback to the patients. This approach is known as self-report dietary data or self-calories log. However many studies pointed out that self-report is not reliable and should not be used as a measure of true energy management [4], because people generally underestimate calories in meals [5], [6] and overestimate calories expended through exercise [7], [8], [9]. Despite

the limitations observed with self-report methods to manage body weight, we cannot exclude this approach in the treatment of overweight and obesity, because they contain valuable, rich, and critical information about calorie intake and expenditure by the population that can be used to inform nutrition policy and assess diet [4]. Supporting self-report dietary data is still important and needed to control body weight, and the research community believes that this support can be achieved by employing technological innovation [10].

1.2 Body Weight Control Strategies and Products

In 2014, a global survey revealed that 50% of world adults have attempted to lose or to control their body weight [11]. Generally, they used various strategies and products including professional-made and/or self-directed diet and exercise plans.

The technology progress of recent years in the area of digital cameras, smartphones and wearable sensors are perceived by researchers and technology-users as an opportunity to conveniently monitor food and exercise. Indeed, 19% of smartphone owners have downloaded one or more health and fitness applications [12], and around 115.4 Million of wearables tracking devices were shipped in 2017 [13].

When it comes to commercially available food tracking and fitness plans, the most common solutions are provided through health and fitness website and smartphone applications (apps). The major dietary and fitness smartphone apps in the market include: LoseIt, MyFitnessPal, FitBit, LiveStrong, and All-in-Fitness. In the research community, diet tracking and monitoring are dominated by systems that rely on smartphone camera and use food image as systems' input. In many studies, food image taken with users' cameras are transmitted to a central server to obtain nutritional and caloric contents of that food. For example, Kawano et al. [14] proposed a mobile food recognition system for estimating calorie and nutrition of foods, as well as recording users' eating habits. The system is a real-time recognition system that runs image recognition techniques on an Android device, using the device embedded camera.

On the other hand, a wide range of ubiquitous and wearable computing systems have been developed by researchers and commercial industries to support

and motivate people into physical exercises. Some of the most popular wearable trackers include Fitbit, Garmin, Recofit, Apple watch, and Samsung. These systems are mainly designed to track physical activity, estimate burned calorie, evaluate user results and support decision-making to improve user performances.

Overall, given the importance of in-and-out calorie estimation for body weight management, various food and exercise tracking systems have been developed to support practitioners, patients, and doctors because the tracking data provide them the sense of direction, help them adjust weight control plans, and enhance the motivation and willingness of any calorie-conscious person to change his/her lifestyle.

1.3 Problem Statement

Despite the existence of many image-based food nutrient calculation systems and research studies, we have to admit that, they generally focus on food identification and classification. Few studies have been done on automated food weight measurement from a single image, because of the challenges of obtaining accurate weight from only one image [15]. Up-to-date image-based food weight estimation systems require the user to take multiple images or use markers, to graphically reconstruct the food in the images. Food weight estimation from a single image still remains an open and challenging problem to overcome for building highly acceptable food calorie and nutrient estimation system; because without knowledge of the food weight, we can not estimate the amount of calorie, needed in the treatment of diet-related diseases [16].

On the other hand, the majority of the wearable and ubiquitous systems for exercises monitoring are either limited to only track aerobic exercise (biking, running) [17], [18], or focus only on muscle and strength training (push-up, chest press) [19], [20]. In other words, each of the existing exercise monitoring systems was designed to only track and evaluate a specific set of physical activities. Thus, the use of such systems tend to limit or constraint the users to perform only within the predefined set of exercises. For example, the system in [21] 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 can not be used for gym exercises which are not performed on a gym mat.

1.4 Research Objectives and Scope

The aim of our research was to use ubiquitous and IoT technologies to propose and develop systems for helping users to easily and efficiently achieve the two main recommendations of the WHO to prevent overweight and obesity:

- Control and measure calorie intake from meals,
- Engage into and evaluate physical exercises.

For both recommendations, our intention was to develop supportive systems that are:

- Easy to use and easy to carry,
- Suitable for daily life,
- Cheap or reasonable price,
- Using everyday life objects and commodity devices.

In this dissertation, we propose two distinct systems to estimate the amount of intake calorie from food and calorie expenditure from exercises. The proposed systems are independent to each other and can be used together or separately.

The first system, which we called NIES (Nutrient Intake Estimation System), is an image-based system that estimates both the food weight and food calorie from a single food picture by using ordinary eating-chopsticks as a measurement reference. This system requires the user to take only a single picture of his/her food, from the top with the chopsticks in the picture. Using several image processing techniques, and meta-data of the food image, NIES automatically estimates the diameter and the height of the food container. Then, given the food type, the system combines the information about the container diameter, height and the food type to provide the weight of the food in the image, and finally estimates the calories of the food.

The second system, called GIFT (Glove for Indoor Fitness Tracking), recognizes and counts physical exercises via a smart-glove. GIFT integrates 16 force sensitive resistor (FSR) sensors into wearable fitness gloves to assess physical exercises, by analyzing the time series of the pressure distribution in the hand palms of the user observed during workout sessions.

Our experimental test on the NIES system ran over 15 different food types achieved an average relative error rate of 6.65% for the food weight measurement and 6.70% relative error for the calorie estimation. The results of the experimental trial on the GIFT system conducted with 10 participants over 10 common fitness exercises showed 88.90% of F-score for overall activity recognition, in the case of user-dependent activity recognition. The result of leave-one-participant-out cross-validation showed 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.

1.5 Dissertation Outline

This dissertation is organized as follows. In Chapter 2, we present existing literature and related works on food calorie measurement and physical activity tracking methods that applied computer visions and/or ubiquitous technologies to evaluate dietary and physical activity. Chapter 3 describes our proposed food weight and calorie estimation method, along with the obtained results and a discussion on the strengths and limitations of this method. Chapter 4 presents the design, implementation and evaluation of the smart-glove to track and assess fitness activity. Chapter 5 concludes this dissertation and provides directions and recommendations for future works.

2. Related Work

In this chapter, we discuss related works on food calorie estimation and physical activity tracking that utilize sensors, computer vision, and IoT technologies to assess dietary and physical activity. The works we review can be grouped into three (3) main categories: food quantity and calorie estimation systems, off-the-shelf exercise trackers, and fitness tracking in-the-lab.

2.1 Food Quantity and Calorie Estimation

2.1.1 Image-based food calorie assessment systems

There has been multiple research on estimating dietary composition of meals using image analysis. For example, Woo et al. [22] and Zhu et al. [15] proposed a Technology-Assisted Dietary Assessment (TADA) system to process food images with a mobile device. In this system, users capture food images using a mobile phone camera. Based on information (i.e., food name and code) determined through food segmentation and classification of the food images, the system chooses a particular food template shape corresponding to the food. And then, the system reconstructs the three-dimensional properties of the food shape by extracting feature points in order to size the food shape template. However, in that system it is assumed that the plate is white, food items in the plate are separated, and users have to take food photos with a chessboard-like marker to calibrate the images; all of these settings are hardly possible in real eating environment. The interesting point of this work is setting users free from any extra operations besides shooting pictures through automated food recognition and 3D volume reconstruction.

As for food recording system with a recognition feature, Food-Log [23], [24] introduced by Aizawa et al., estimates food balance by dividing a received food image into 300 blocks. For each blocks, it extracts the color and DCT coefficients, and classifies into five groups such as staple, main dish, side dish, fruit, and non-food. Yang et al. [25] proposed a method to estimate the nutrition in foods. They calculate pairwise statistics between local features computed over pixel-level segmentation of the food image into eight ingredient types. The accuracy of this system was up to 28%.

As it appears in the above-mentioned systems, the works are mainly focused on the identification of the "menu" or the "food item", without much consideration of the food volume or the food weight in their estimation. Our proposed system can estimate the food weight with no marker required. Note that our current system estimates only food weight and not food types, and it requires user 's assistance to enter the food name.

2.1.2 Food quantity estimation systems

Food portion size estimation is extremely difficult since foods shapes and appearances are subject to variations due to the cooking circumstances and eating conditions. For this reason, food volume and weight estimation for dietary management have been a challenge for ubiquitous computing and healthcare systems. Most image-based food volume estimation systems use multiple images [26], [27], video [28] or 3D reconstruction [29]. For example, Kong et al. developed an application called DietCam [30] that automatically assesses food intake based on multiple views of food. The user is required to take three pictures of the same food separated by 120 degrees, in order to get the food volume. This requirement increases the burden on the user.

Another approach appears in [31] where two pictures must be taken, one from the top and one from the side, with the user 's thumb placed beside the dish when taking the picture from the top. In this study, they explored three categories of food: single food, non-mixed food, and mixed food. While this system showed good results for single and non-mixed foods such as eggs, oranges, and apples, it had problems with mixed foods such as soup and curry.

Chen et al. [32] introduced a 3D/2D model-based image registration method for quantitative food intake assessment. Their method uses global contours to determine the position, orientation, and scale of the user-selected 3D shape model. The volume estimation obtained is accurate for food items such as oranges or hamburgers with a simple model. However for food items such as bananas or salads that have a complex model, this system does not have a solution. There are also methods utilizing sensorized eating surfaces such as smart tables equipped with sensors, or modified tray to weigh the food during the meal time. For example,

Table 1: Comparison to some related food tracking works

	Require special equipment	**Burden level**	**Place restriction**	**Volume or weight estimation**	**Accuracy**
DietCam [30]	No	High	No	Yes	Medium
Foodlog [24]	No	Low	No	No	Low
Smart surface [33]	Yes	Low	Yes	Yes	Good
Our system	No	Low	No	Yes	Good

Bo et al. [33] proposed a smart tablecloth equipped with a fine-grained textile matrix and a weight sensitive tablet. Based on user actions such as cutting or stirring, they determined the food type, and by using pressure sensor under the dining tray, they measured the food weight. However, this requires the user to eat only on that table-cloth and dining tray. Also the system does not allow user to move or change the position of the plate. Such restrictions are impractical, if we want to use this system in an ordinary eating-place such as restaurant or food-corner.

Compared to the above methods, our proposed food calorie estimation system is less obtrusive, less burdensome, suitable for any eating-environment and requires only one food picture for food weight and calorie measurement. Table 1 compares some of the related works to our proposed food calorie estimation system.

2.2 Off-the-shelf Exercise Trackers

In this section, we focus on physical activity trackers that are already on the market.
Commercially available devices such as running watches, fitness wristbands or

trackers are nowadays ubiquitous tools used by thousands of people. Fitbit [34] and Garmin [35] 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-up), and it requires the user to manually log strength training such as dumbbell curl [36]. Another well-known commercial system is ActiveLinxx (formerly known as Fitlinxx) [37]. 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 users' smartphone or tablet, this system still requires the fitness center owners to purchase and install these training machines.
GoogleFit 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 built-in sensors of that device to track commonly-measured fitness data such as the number of steps and calories burned. However, GoogleFit's support website revealed some limitations of the system [38]. For example, it is reported that walking, running and biking detection is not perfect on GoogleFit, because only smartphone or tablet sensors are not accurate enough for tracking this type of activities.

2.3 Fitness Tracking in-the-lab

2.3.1 Glove-based systems

As for research works that use gloves for tracking physical activities, Chang et al. [39] 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 bench press, biceps curl, and lateral raise.
Another work presented by Qiang et al. [40] 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 Flexi-Force) attached onside each glove and connected to a micro-controller 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.

2.3.2 Pressure sensing based systems used in fitness activity

Sundholm et al. [21] 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 kNN-classifier and counting accuracy of 89.9%. The major limitation of this system is that it can not 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 analyses hand palm interactions even for exercises that are not performed on a mat.
A different work introduced by Zhou et al. in [41] 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. Bo et al. [42] made a smart soccer shoe that uses pressure sensing matrices to detect and analyze the interaction between players 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.

2.3.3 Other sensor-based fitness systems

A state-of-art study, similar to our work is the "Recofit" system developed by Microsoft Research [19]. Recofit uses inertial sensors attached to the upper forearm of the athlete and measures 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, pushup, shoulder press, triceps extension, back fly, etc. Although this result is impressive, there are several points to remark for comparison to our work. First, Recofit uses only auto-correlation 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 (flexibility training, dynamic strength training, static strength training and circuit training), while the Recofit system focuses on 2 exercise groups which are weight training and calisthenics.
Another more recent work developed by M. Hassan et. al [17] 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 term of using FSR sensors, it was mainly designed to detect a running strike and provide real-time feedback without recognizing and/or counting 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 to re-adjust existing layouts of fitness centers or buy new fitness machines. Table

2 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.

Table 2: Comparison to some related activity tracking works

Related systems denomination	**Targeted exercises**	**Type of sensors used**	**Collected data type**	**Classification / Counting**
Tracking Free-Weight Exercises [39]	Weight training exercises	3-axis accelerometers	Acceleration	Yes / Yes
Force-Sensing Glove System for Motorbike Riding [40]	Motorbike riding	Flexi-force tactile sensors	Tactile forces	Yes / No
Smart-Mat [21]	Strength and muscular endurance	Resistive pressure sensor matrix	a) Body weight distribution b) Pressure distribution	Yes / Yes
Never Skip Leg Day [41]	Leg Exercises	Textile FSR sensors	Pressure distribution	Yes / Yes
Recofit [19]	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

2.4 Chapter Summary

Ubiquitous and IoT-based systems for tracking and assessing both the dietary and physical activity have been widely studied in the past few years. However most of these systems showed some limitations especially the lack of volume and weight estimation of the eating food in the case of diet tracking, and the limited range/number of tracked exercises in the case of physical activity tracking. In the following chapters of this dissertation, we present the details of our two (2) ubiquitous systems in the attempt to push the limits of existing dietary and physical activity tracking systems. NIES (Nutrient Intake Estimation System) is our proposed system for diet tracking and food calorie estimation, and GIFT (Glove for Indoor Fitness Tracking) is our proposed IoT-based system to recognize, count and track physical exercises.

Figure 1 gives an overview of the relation between existing systems and our proposed systems, along with the foundations of this thesis.

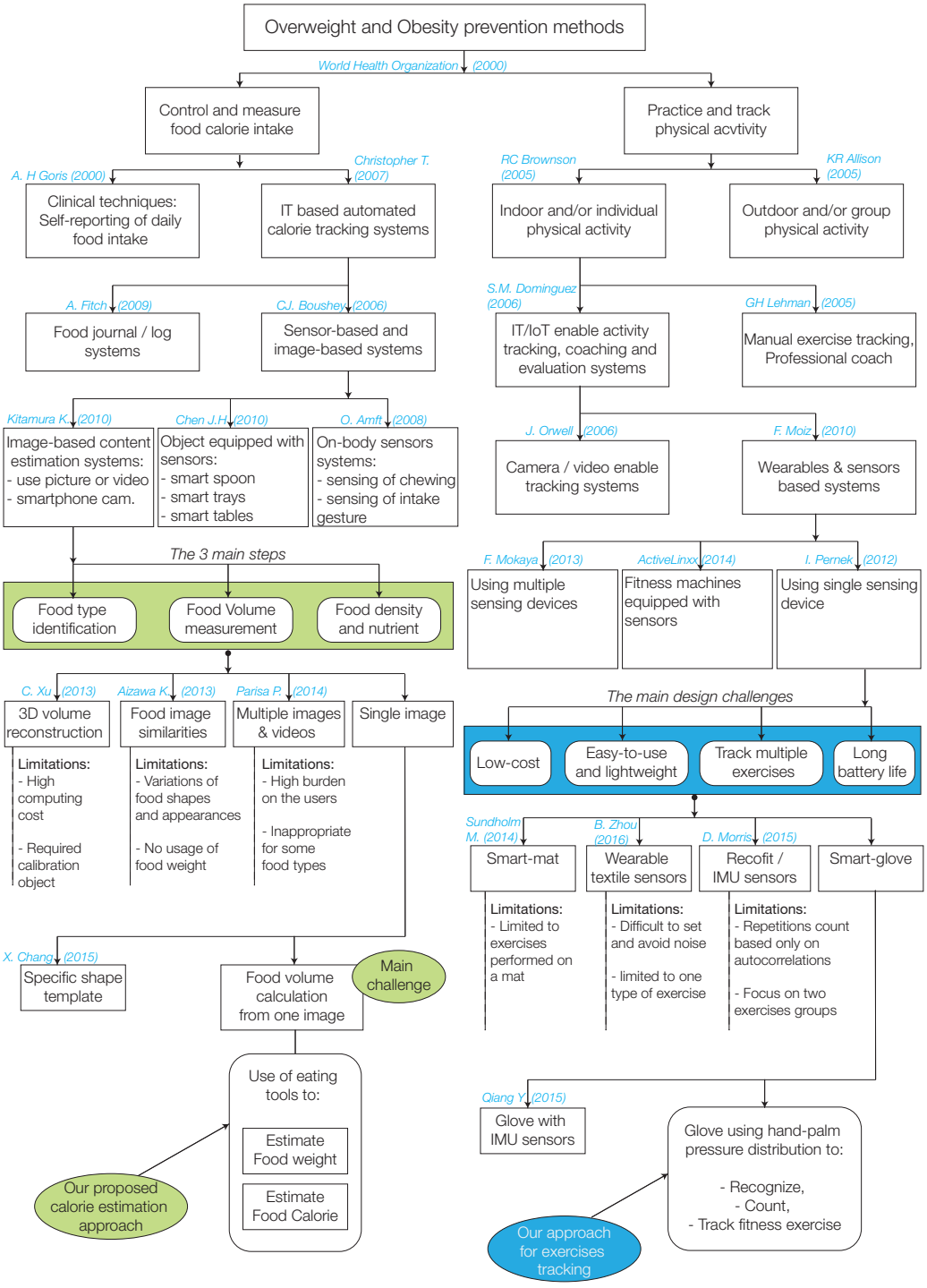


Figure 1: Foundation of the proposed systems and relation to previous works

3. NIES: Nutrient Intake Estimation System

3.1 Introduction

As described previously in Section 1.3, despite the existence of many image-based food nutrient calculation systems and research studies, food weight estimation from a single image still remains an open and challenging problem to overcome; because without knowledge of the food weight, one's can not concretely estimate the amount of calorie in a food. In this Chapter, we present our novel approach to estimate food weight and calorie from a single image, with the help of eating tools especially chopsticks which are used during meal times.

The proposed system just requires the user to take a single image from the top with chopsticks visible in the image as shown in Fig. 2, with one stick positioned on the edge of the container, while the other stick is put on the table close to the container. For our experiment, we focused on bowl-shape containers and utilize chopsticks of length 22 cm of our university's restaurant, as measure reference, and apply computer vision techniques on the food picture, which contains the chopsticks. The fact that the system uses daily-life chopsticks, that can be found everywhere, makes the system pervasive and suppress the burden of always carrying calibration objects as observed in existing systems like in [43].

Via a smartphone application, the user sends the taken image to the server. At the server side, the system automatically measures the diameter and the height of the food container using several images processing techniques, the camera's focal length and sensor size contained in the exchangeable image file (EXIF) metadata of the image. Then, given the food type, the system combines the information about the container diameter, the height and the food type to provide the weight of the food in the image, and finally estimates the calories of the food. Our proposed method aims to estimate a calorie of a food by:

- calculating the food weight from a single shot image,
- estimating the calorie from the calculated weight using a weight to calorie nutrient database tables.

We present two contributions to the research field of image-based calorie estimation:

- Usage of eating tools as measurement reference, in replacement of the usually use calibration cards,
- A calorie estimation method that considers foods served in containers having bowl shape.

Figure 2: Food image with chopsticks on the top and at the bottom of a container.

3.2 Food Weight Measurement Method

One goal of our work is to help users record daily food as well as obtaining an accurate weight estimation from a single picture of the food taken before eating. An important part of our weight measurement process is the use of the chopsticks present in the food image.

Chopsticks, which can be found easily in any kind of eating environments such as homes, restaurants, and vending machines, are used as a reference for measurement. We take advantage of the chopstick's industry standards and well known-length (20 cm to 22 cm) to compute the volume and weight of the foods in the images.

In this work, we utilized disposable chopsticks measuring 22 cm, which are the most commonly used and available in the university's restaurant. We also used the fact that generally foods are served in standardized food containers, designed

with regular shapes and normalized dimensions. Thanks to this standard, the Japanese curry rice, for example, is rarely served in containers for ramen (Chinese noodle).
In the early development stage of our system, we used chopsticks as reference measurement to only estimate the diameter of the container in the picture and then derive the height of the container from a pre-built diameter-to-height table, which matches the diameter to the height based on the food type. However, this approach was highly dependent on the accuracy of the pre-built diameter-to-height table, and the obtained result was not satisfactory enough. Thereupon, we proposed a weight estimation method that computes the diameter and the height of the food container from a single image of the food, using several computer vision techniques combined with the EXIF metadata of the image [44]. With the food name provided, the system is able to estimate the food volume. Then, we used the food volume and a density table to extract the mass of the food, and finally estimate the food caloric content. In the following sections, we will firstly explain how we compute the diameter of the food container through image processing. Secondly, we will demonstrate how to get the container height from a single image, using chopsticks and finally, we will explain how we combine the food type, density table values and container sizes, to obtain the food volume and its weight.

3.2.1 Measurement of the Food Container Diameter

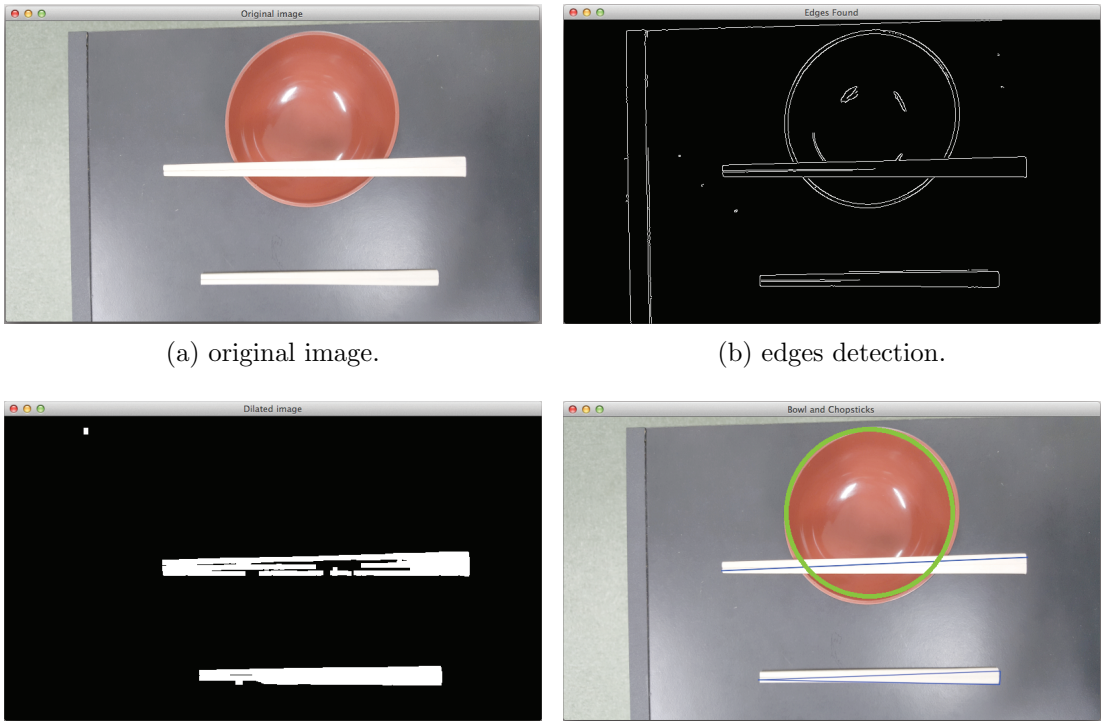
To measure the diameter of the food container from the food picture, we need to know the geometric shape and the size (pixels) of the container and the length of the chopsticks (in pixels) in the picture.

Container shape detection

For the geometric shape problem, since foods are generally served in round shape containers, we decided to initially detect the circle shape of the containers open-top. To detect the circle shape of round dishes, we use a computer vision feature extraction technique called the Hough transform.
The Hough transform can be used to isolate features of a particular shape within an image. Because it requires the desired features to be specified in some paramet-

ric form, the classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. [45]. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.
The Hough transform has two essential parameters, m and n. The parameter "m" is used for the Canny edge detector, while "n" is for the center detection stage [46]. The Canny operator is an optimal edge detector that takes as input a gray scale image, and produces as output an image showing the positions of tracked intensity discontinuities, which are the edges of the input image. The number of circles detected by the Hough transform depends on these two parameters. To detect the circle of the main round dish with high precision, after several tests, we fixed these two parameters to 10 and 100. The smaller n is, the more false circles may be detected. However, with n fixed to 100, we always obtained good circle detection with all the tested food images. The result of the Hough transform is the circle formed by the round shape of the container and its radius in pixels.
We store this value of the radius, which will be used later with the result of the next subsection, to find the diameter of the container in real life.

Determination of the chopsticks length in the image
The next step consists of getting the length of the chopsticks in the image. We achieve this step by applying the Canny edge algorithm to find the edges of the input image (Fig. 5). The Canny edge requires three parameters: low threshold, high threshold, and the kernel size.
The low threshold is used for edge linking, while the high threshold is used to find a segment with strong edges. We previously presented the parameter "m". This parameter represents the low threshold that the Canny edge uses to link the edges. If no value is given to the high threshold, the value of the high threshold is, by default, set to three times the value of the low threshold, according to Canny's recommendation [47].
The kernel size is the size of the Sobel operator to be used inside. The Sobel operator is an algorithm which emphasizes regions of high spatial frequency that correspond to edges.
The output obtained from the Canny algorithm is processed to find contours in



(a) original image.

(b) edges detection.

(c) thresholded image.

(d) output image.

Figure 3: Processing steps for container diameter and chopsticks length measurement.

the food image. From the possible contours found, we only extract rectangular contours and then retrieve the length in pixels of the chopsticks by taking the length of the stick placed on the container, which appears as the longest stick in the food image (Fig. 5).

Finally, all the obtained results, the container' radius in the picture, the chopsticks length in the image and the known length of the chopsticks in real life, are combined in a cross multiplication to determine the diameter of the container in real life.

Table 3 shows the results of our diameter estimation method applied to 25 images of 5 different containers' types (5 images for each type of container). The pictures were taken at different positions but from the top, and the 25 containers were empty. We decide to run the first test on empty containers to evaluate the

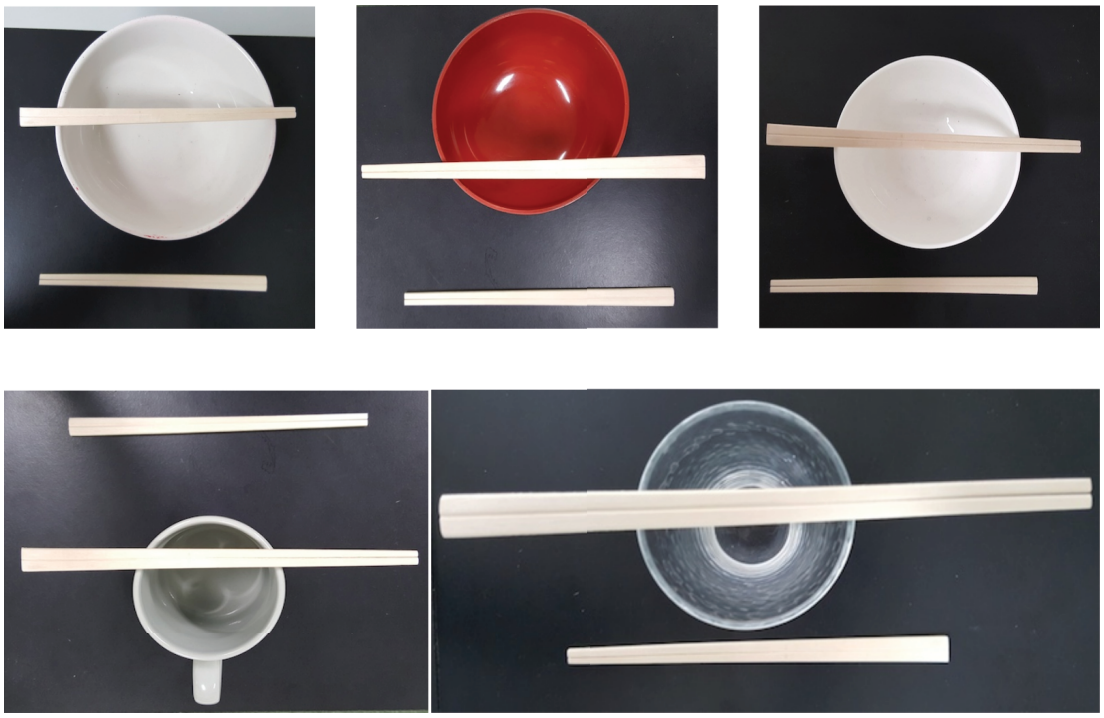


Figure 4: Types of bowl used for testing

accuracy of the proposed method before testing the system with non-empty containers. The average estimation, the standard deviation (SD), and the relative error of each estimation are also presented in the table.
Figure 4 shows the five different types of bowl used for testing. "Small bowl" refers to bowl type used to serve white rice or salad, while "Big bowl" are for udon, oyakodon, ramen, etc. The "miso soup" bowl is used for serving of miso paste soup and other kinds of soup. The other two bowl types' names are self-explanatory (coffee, tea and water).

As shown in the table, the diameter estimation method achieves, in the best case, a relative error rate of 3.75%, and 8.75% in the worst case, with the "coffee cup". For most of the container types, we obtained a standard deviation less than 0.9, except for the "Big bowl."

Table 3: Container diameter estimation

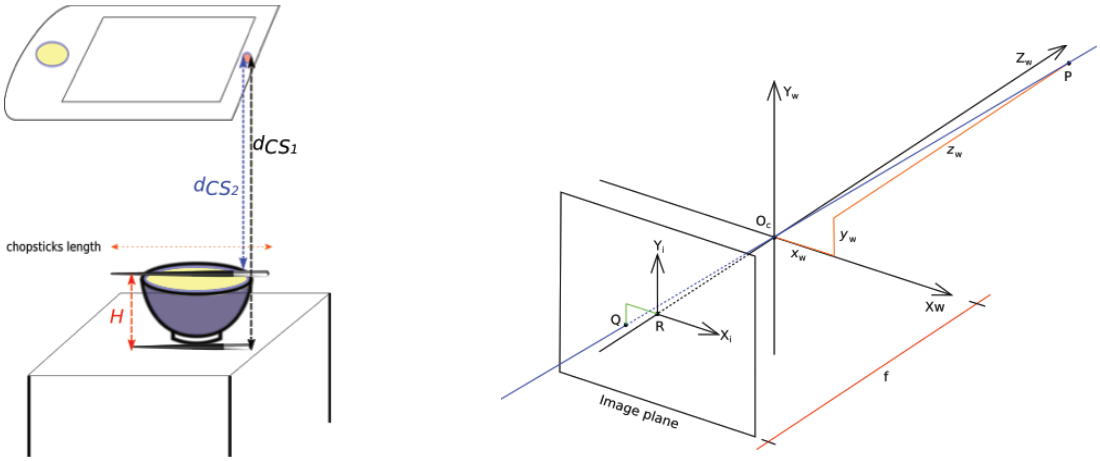
	Miso bowl	Small bowl	Big bowl	coffee or tea cup	water or juice glass
Real diameter (cm)	**12**	**12**	**16**	**8**	**7**
Picture 1	11	12	14	7	6
Picture 2	11	11	16	8	6
Picture 3	12	11	15	6.5	7
Picture 4	12	12	17	8	7
Picture 5	10	11	15	7	6
Average Estimation	**11.20**	**11.40**	**15.40**	**7.30**	**6.40**
Standard Deviation	**0.83**	**0.57**	**1.14**	**0.70**	**0.54**
Relative Error (%)	**6.67**	**5.00**	**3.75**	**8.75**	**8.57**

3.2.2 Measurement of the Food Container Height

To obtain the food volume, we need to get not only the container diameter but also the container height. In our estimation method, we assume that the container height is nearly equal to the distance between the two sticks of the chopsticks, when one stick is on the containers' top and the other placed at the bottom near the container (Fig. 5a). This actual height can appear differently on the food image, due to the perspective distortion produced by the smartphone camera lens when taking the photo. In our current system, we assumed the "ideal case" in which the distortion created by the camera lens does not affect excessively the container height on the image. Our proposed method does not undistort the food images.

We get the distance between the two sticks by using the triangle similarity theorem applied to the pinhole camera model explained in the next paragraph. The container height is then calculated by:

$$H = d_{CS_1} - d_{CS_2} \tag{1}$$



(a) Container height measurement.

(b) Pinhole camera model.

Figure 5: Measurement of the Food Container Height.

where d_{CS_1} and d_{CS_2} are respectively the distances from the camera to the stick 1 and the stick 2 (Fig. 5a).

Pinhole camera model

The pinhole model describes the geometrical and mathematical relationship between a camera image plane and the objects in space. Figure 5b shows the pinhole model used in our proposed measurement method.

A point P in world coordinate (X_w,Y_w,Z_w) has a 3D coordinate of (x_w, y_w, z_w). The line (blue line) of the point P passes through the point P and the point O_c , which is the origin of the camera coordinate system(X_c,Y_c,Z_c). The image plane, where 3D points are projected through the camera, is located at distance f (camera focal length) from the origin of the camera. This image plane is perpendicular to the Z_w axis, which is the optical axis of the camera. This optical axis points is the viewing direction of the camera. The point R, referred as the image center, corresponds to the intersection of the image plane and the optical axis. The projection of the 3D point onto the 2D image plane is the point Q. This point lies at the intersection of the image plane and the projection line. The point Q has coordinates (x_i, y_i) in the 2D coordinate system (X_i,Y_i) with the origin R of the image plane.

To extract the distance from the camera to the point P, we need to find the mathematical relation between the 2D coordinates of the point Q (x_i, y_i) and the 3D coordinates (x_w, y_w, z_w) of the point P.

Triangle similarity

Two geometrical objects are similar if each object is congruent to the result of a particular uniform scaling of the other object. Figure 6, which represents the same scene as Fig. 5b, but from above, looking down in the negative direction of Y_w axis, possess two similar triangles O_cfQ and O_cZ_wP.

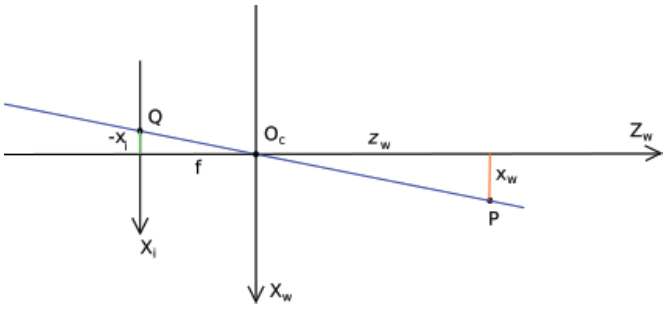


Figure 6: Pinhole camera model seen from Y_w

Both triangles have part of the projection line as hypotenuses. The two adjacent sides of the left triangle to the right angle are $-x_i$ and f (focal length of the camera). For the right triangle, the adjacent sides are x_w and z_w. Since these two triangles are similar, we get the following equation:

$$\frac{-x_i}{f} = \frac{x_w}{z_w} \Rightarrow x_i = -\frac{f}{z_w} x_w. \tag{2}$$

In the same way, when looking in the negative direction of X_w axis of the world coordinate, we get:

$$\frac{-y_i}{f} = \frac{y_w}{z_w} \Rightarrow y_i = -\frac{f}{z_w} y_w. \tag{3}$$

From these two equations, the relation between (x_i, y_i) and (x_w, y_w, z_w) is derived:

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = -\frac{f}{z_w} \begin{pmatrix} x_w \\ y_w \end{pmatrix}. \tag{4}$$

Because the mapping of a point from the world coordinate to a 2D coordinate described in the pinhole model is a perspective projection followed by a 180 degree (180°) in the image plane, the resulting image is rotated by 180 degree. In order to get the expected image from the camera, we need to rotate the coordinate system in the image plan by 180 degree. Finally, after rotation, the mapping from 3D coordinate of a point P (x_w, y_w, z_w) to the 2D coordinate point Q (x_i, y_i) is given by:

$$x_i = \frac{f}{z_w} x_w \qquad and \qquad y_i = \frac{f}{z_w} y_w. \tag{5}$$

Measurement of the distance from camera to stick

To find the distance from a camera to an object or a marker, computer vision techniques exploit the triangle similarity and pinhole model results, described in the two previous paragraphs. The camera to object distance is given by:

$$\frac{P}{F} = \frac{W}{Dist_{cam\_obj}} \Rightarrow Dist_{cam\_obj} = \frac{F \cdot W}{P}, \tag{6}$$

where F (mm) is the focal length of the camera, W (mm) the known width of the object or marker, and P (pixels) the apparent width of the object in the image. Given the sensor size of the camera, the formula can be defined as:

$$Dist_{cam\_obj} = \frac{F \cdot W \cdot I_w}{O_w \cdot S_z}, \tag{7}$$

where I_w is the image width, O_w the object width (pixels) and S_z the sensor size (mm).

In our current implementation, we use the food images from smartphone, and for each image we extract the EXIF data of the camera system, which contains metadata such as the camera focal length and the sensor size. In Section 3.2.1, we explained how to get the width of each stick of the chopsticks. We input these values into (7) to determine the distance d_{CS_1} and d_{CS_2} from the camera to each

Table 4: Container height estimation

	Miso bowl	Small bowl	Big bowl	coffee or tea cup	water or juice glass
Real height (cm)	**7.5**	**6.5**	**8**	**10**	**12**
Picture 1	8	6	7	13	12
Picture 2	7	6	9	11	13
Picture 3	6	5	8.5	10.7	10
Picture 4	9	6	7	10	15
Picture 5	8	6	6.5	9.5	14
Average Estimation	**7.6**	**5.75**	**7.6**	**10.84**	**12.8**
Standard Deviation	**1.14**	**0.5**	**1.08**	**1.34**	**1.92**
Relative Error (%)	**1.33**	**11.54**	**5.00**	**8.40**	**6.67**

stick. Finally, we get the height H of the food container by taking the difference between d_{CS_1} and d_{CS_2} (see (1)).

Table 4 shows the results of applying our height measurement method to the same 25 images of empty containers as described in Section 3.2.1. The table also gives the average estimation, the standard deviation (SD) along with the relative error of the measurement for each container.

As shown in the table, the height estimation method achieves a relative error rate of 1.33% in the best case with the "Miso bowl," and 11.54% in the worst case, with the " Small bowl."

3.2.3 Volume Estimation and Food Type Identification

Estimation of the food volume is a critical step for any calorie estimation system for dietary assessment. In our proposed system, after determining the container dimensions, we estimate the food volume from the food image by firstly estimating the container volume, and then using the density information for that particular

food, we estimate its volume and weight. Currently, we use the "aqua-calc" table [48] which provides a volume to mass conversion for more than 4000 food items and ingredients.

Volume estimation

In our current implementation of our NIES, we focus on the foods that are served in containers of bowl shape. A bowl is a round, open-top container used in many cultures, especially in Japanese culture, to serve foods. Geometrically, classic bowl can be represented as spherical cap. A spherical cap is a section of a sphere slide off by a plane (fig.7).

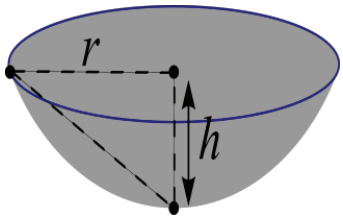


Figure 7: Spherical cap

The volume of a spherical cap is defined by this ordinary used equation:

$$V = \frac{\pi}{6} \cdot h \cdot 3r^2 + h^2, \tag{8}$$

where r is the radius of the base of the cap, and h its height. These two parameters are obtained from the methods described in the Sections 3.2.1 and 3.2.2.

Figure 8 shows the results of our volume estimation using (8). The best volume estimation are obtained with the "Water or juice glass", and the"coffee or tea cup" with respectively a percentage error rate of 1.38% and 1.98%. The system produces acceptable results for other container types.

Knowing the container volume, which we assume to be closely equal to the food volume when the bowl is full, we can get the food weight using density table information of each food type.

Food type identification

Food recognition and classification has been the focus of many image-processing

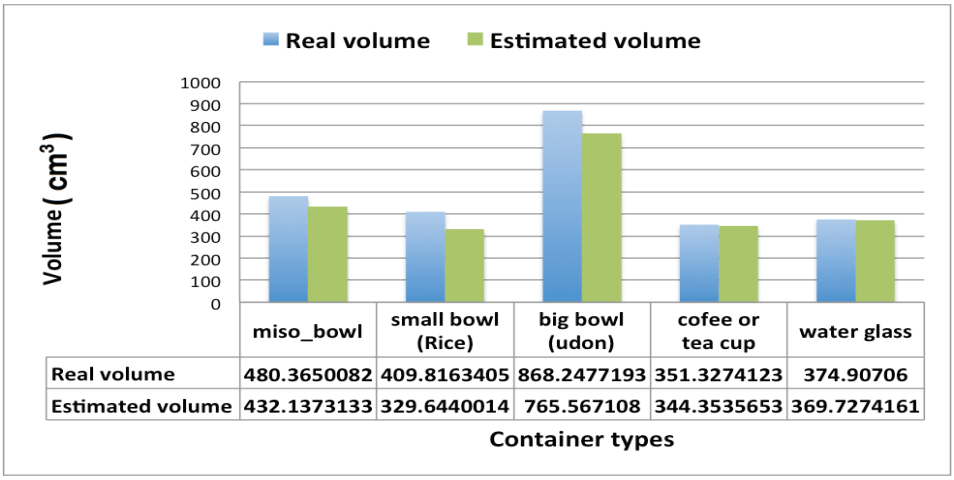


Figure 8: Volume estimation performance

studies in the recent years. Among these studies, some exploited the local features of food images to identify the food type [25], while other focused on global features such as global histogram [49]. Other studies utilized features extraction techniques such as scale invariant features transform (SIFT) [50] for food image classification. Kagaya et al. proposed a food detection and recognition system using a deep-learning approach called convolutional neural network (CNN) [51]. However in most cases, the accuracy of the classifiers proposed in these study are questionable and/or require further improvements to perform as expected.
In our current implementation, we did not implement any food classification technique; we manually enter the food name in the system.
As mentioned before, the most critical and challenging step in dietary assessment is the food weight estimation from the food image. Therefore, at this stage, we assume that the food type has been well identified or given manually.

Density table

Food density tables are databases which provide a tool for researchers and professionals of food analysis to convert volume into weight and vice-versa. Data collected are prepared from the literature, various national food composition tables and measurements are conducted by international and national organization such as FAO (Food and Agriculture Organization of the United Nations), FNDDS

Table 5: Density table for some food types

Food name and description	**Density** (g/cm^3)	**Volume** (cm^3)	**Weight**(g)
Bread, dry mix, prepared	0.845	150	126.75
Natto	0.74	150	111
Noodles, Chinese chow mein	0.19	150	28.5
Rice, white, raw	0.782	150	117.3
Soybeans, green, raw	1.082	150	162.3
Tofu, raw, regular	1.048	150	157.2
Tomato sauce	1.036	150	155.5
Orange juice, raw	1.048	150	157.2
Egg, white, raw, fresh	1.027	150	154.05
Apple juice, unsweetened	1.048	150	157.2

(Food and Nutrient Database for Dietary Studies) or USDA (United States Department of Agriculture).
In this study, we used the "aqua-calc" density [48], since it has data available for Japanese food types as well as international food types. It provides a volume to mass conversion for more than 4000 food items and ingredients. Table 5 shows for some foods, their density along with the conversion to weight for a volume of 150 cm^3 of each food.
In our system, the food weight is obtained by multiplying the estimated volume acquired from the food image with the specific food density as shown in (9).

$$food\ weight = estimated\ volume \times food\ density. \tag{9}$$

3.3 Food Journaling and Experiment

To validate the measurement obtained from our proposed estimation system, we performed validation experiments using images of various food types (rice, ramen, miso, oyakodon, guydon) served in the five container types mentioned

before. Over three weeks, we collected the food images from 15 participants (12 male) aged 23-31, through an Android food journal application (app) that we developed. To certify the accuracy of our system, from the collected images, we selected only images that are from foods served in our university restaurant, since we know their pre-estimated weight and calorie. The details about this food journal app and the results of our experiment are discussed in the following sections.

3.3.1 Image Collection through Smartphone Application

One mission of the proposed system is to help people who suffers from obesity, overweight and diet-related disease to keep a record of the amount of daily nutrients they consumed without the need for recording this data manually. The functions of our system are to calculate not only the weight of the food in the picture, but also the amount of calories and to keep a food log of the users' consumed foods. To accomplish this, we developed a smartphone app that runs on Android devices and takes advantage of the built-in camera, processing power and network capabilities of the smartphones. The app represents the interface for users to send and receive data about their foods. Over three weeks, the participants used this app to collect food images for testing the estimation method proposed in this study. The application is also utilized in another research study, in which we assess the consistency of meal-tracking and behavior change using a game-based approach [52]. Figure 9 shows some screenshots of the application relevant to the food images collection and the proposed estimation method.
In the "Welcome screen," users have the control buttons to navigate to the game screens, upload screens and to other options screens such as "food log-calendar" and pop-ups. The "Upload screen" allows users to take a photo or choose a photo from the gallery and contains a form where users can fill in additional information such as hunger level and mood. Figure 10 shows some of the images received during the 3-week food image collection campaign.

(a) Welcome screen. (b) Upload screen. (c) Recommendation screen.

Figure 9: Smartphone application for food images collection.

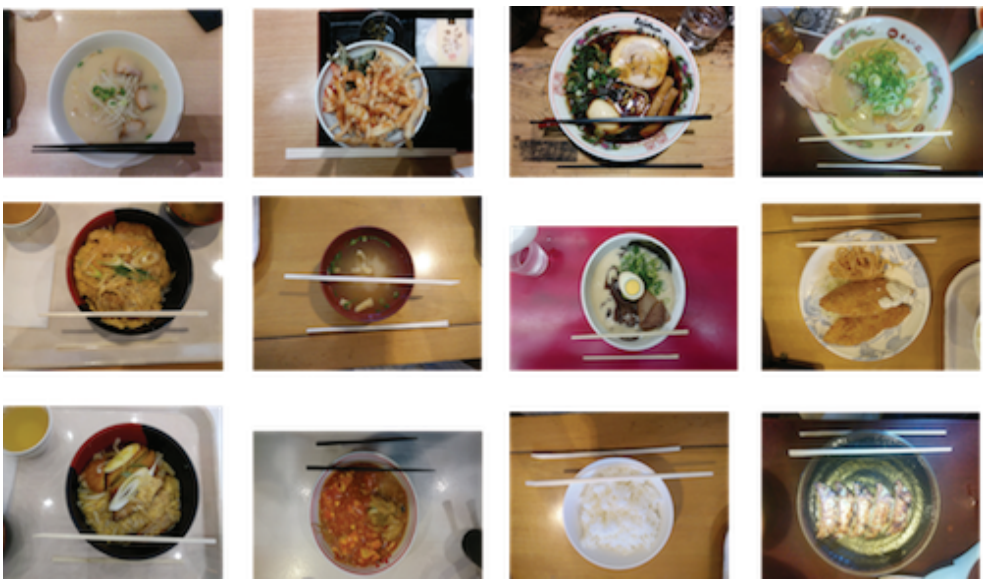


Figure 10: Examples of food images collected

3.3.2 Results of the Weight Measurement

During the period of food images collection, we received 119 images from the participants. Among these images, about 40 images were not taken from top or with the chopsticks not positioned as shown in Fig.2.
Some other images were not foods served in round dish containers. These kinds of images were initially accepted because the image collection application is used for a separate study, which does not have restrictions on the food photo, but we do not include them in this evaluation. Figure 11 shows some of the food images received but not used in the experiment. Finally we worked with 50 images, grouped into 15 food types. From Equation (9), we compute the weight of these foods images.

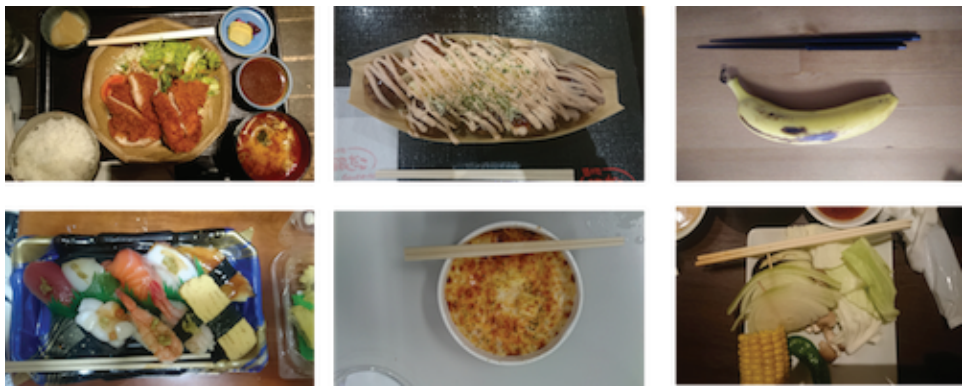


Figure 11: Examples of food images not used

Table 6 shows for each food type the real weight (average weight), the estimated weight, the difference between these two values, and the relative error. The lowest relative error 3.60% was obtained from the estimation of the katsudon, and the highest rate came from the big bowl of rice with 10.17%. About 13 of the food types have their relative errors less than 10%.
The table also shows the differences of the estimated weight to the real measurements for each food images. The differences show that 8 food types (rice, ramen, miso, guydon, fried rice, tendon, kakesoba, and tonjiru) were overestimated, while the other types were underestimated. The largest overestimation

was obtained from the gyudon weight estimation, with a difference of 33.8g. The average relative error for the weight estimation is about 6.78%.

Table 6: Results of proposed method in comparison with real values

Food type	**Average measured weight (g)**	**Estimated weight (g)**	**Absolute Error (g)**	**Relative Error (%)**
Oyakodon	468.5	450.5	-18	3.84
Tamagodon	444.5	422	-22.5	5.06
Katsudon	413.6	398.7	-14.9	3.60
Kakeudon	405	398.5	-6.5	1.60
Rice (small size)	160	174.5	14.5	9.06
Rice (regular size)	340.4	305.4	-35	10.28
Ramen	650	680	30	4.62
Miso soup	143	155	12	8.39
Gyudon	416.5	450.3	33.8	8.12
Fried rice	371.8	400	28.2	7.58
Tendon	380	400	20	5.26
Kake soba	375	402	27	7.20
Kitsunu soba	428	385.5	-42.5	9.93
Tsukimi soba	434	390	-44	10.14
Tonjiru	176	185	9	5.11
Average relative error				**6.65%**

3.3.3 Calorie Estimation

Once we know the estimated food weight, we can move on to the final step of estimating the amount of calorie (energy) of the food in the image. To achieve this step, we rely on nutritional fact database (NFD) and food guide as reference to provide nutritional information. Previous studies show that NFD and food

guide are important components to realize useful and successful calorie estimation system [28]. Details about nutritional values of various food types are stored in these databases and are available from national and international health organizations.

In Japan, the Ministry of Health, Labour and Welfare (MHLW) and the Ministry of Agriculture, Forestry and Fisheries (MAFF) introduced the "Spinning food guide" as the food and nutrition reference tool to help people practice healthy eating. Figure 12 (Source: Ministry of Agriculture, Forestry and Fisheries website [53]) shows the Japanese spinning food guide accompanied by a chart that indicates the recommended daily servings for each food group.

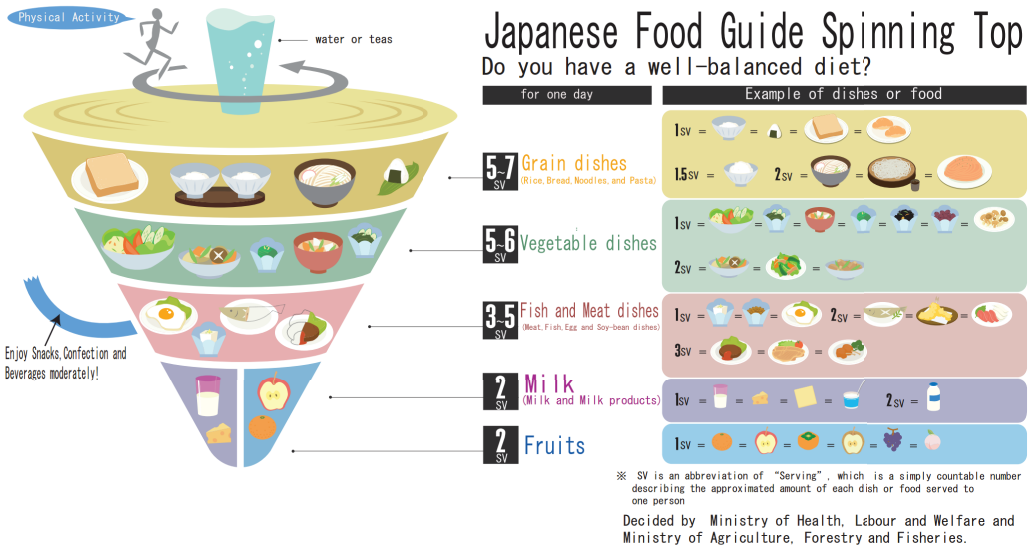


Figure 12: Japanese food guide spinning top

Our system currently uses such databases as reference to estimate the amount of calorie of the food in the received images.

For this study, since we used images of food served in our institute's restaurant, we adopted the restaurant pre-estimated food calorie database as ground truth data to evaluate our proposed system.

Table 7 shows for each food type the real values of weight and calorie, followed

by the estimated weight and calories. The absolute error and relative error between each estimated value and the real values are also presented in the table. A quick observation of Tables 6 and 7 reveal that the values of the relative error obtained during the weight measurement are really close to the relative error values achieved in the calorie estimation. This observation allows us to confirm that the food calorie estimation strongly depends on the food weight measurement. The average relative error for the calorie estimation is about 6.70%. The worst case of underestimation is observed from the Rice (regular size) with an absolute error of -59 kcal, while the worst overestimated food shown an absolute error of $+63$ kcal (guydon). According to the USDA nutrient Database [54], 63 kcal represent the quantity of eneregy (calorie) in 1 medium raw egg of 44g. From this, we can assert that our proposed system has an overall acceptable result, and can work better if we overcome some current limitations, described in the next section.

Table 7: Result of the food calorie estimation

		Real values		**Estimated values**		**Calorie estimation error rates**	
N	**Food Type**	**Weight (g)**	**Calorie (kcal)**	**Weight (g)**	**Calorie (kcal)**	**Absolute Error (kcal)**	**Relative Error (%)**
1	Oyakodon	468.5	684	450.5	658	-26	3.80
2	Tamagodon	444.5	636	422	604	-32	5.03
3	Katsudon	413.6	823	398.7	794	-29	3.52
4	Kakeudon	405	320	398.5	315	-5	1.56
5	Rice (small size)	160	269	174.5	294	+25	9.29
6	Rice (regular size)	340.4	572	305.4	513	-59	10.31
7	Ramen	650	553	680	578	+25	4.52
8	Miso soup	143	46	155	50	+4	8.70
9	Gyudon	416.5	771	450.3	834	+63	8.17
10	Fried rice	371.8	673	400	724	+51	7.58
11	Tendon	380	642	400	676	+34	5.30
12	Kakesoba	375	304	402	326	+22	7.24
13	Kitsunu soba	428	441	385.5	398	-43	9.75
14	Tsukimi soba	434	378	390	340	-38	10.05
15	Tonjiru	176	121	185	128	+7	5.79
Average relative error							**6.70%**

3.4 Discussion

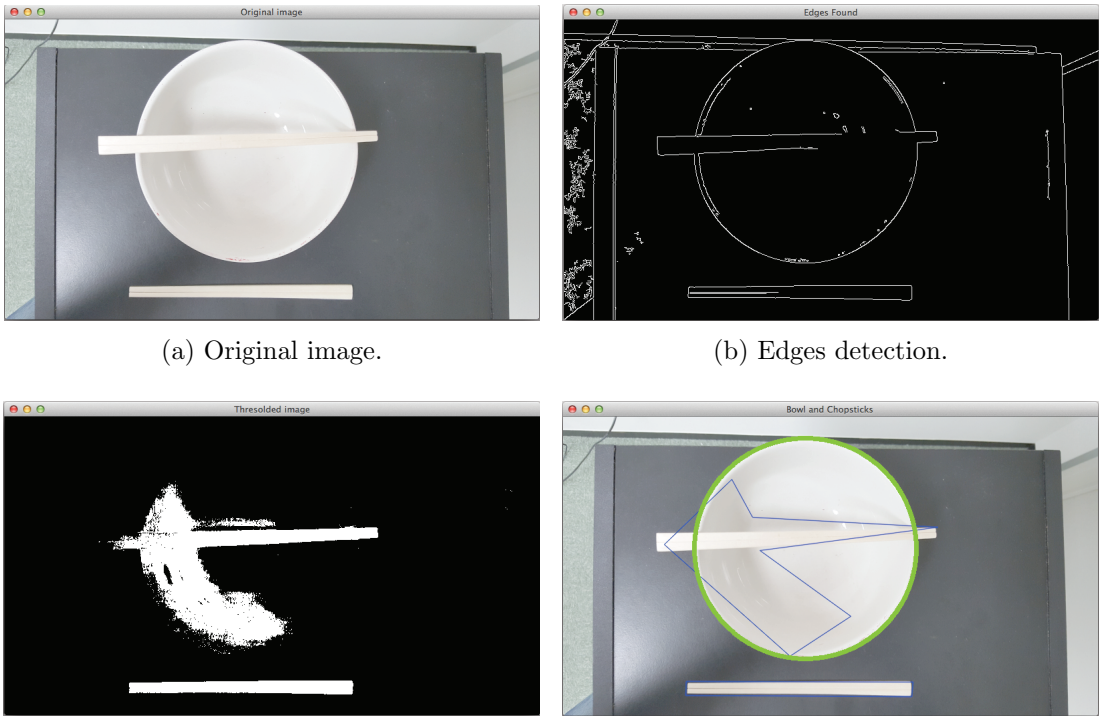
Our experiment revealed several strengths and weaknesses in our proposed calorie estimation system. In the rest of this section, we discuss some of the strengths, limitations and how we plan to overcome them in our future work.

3.4.1 Food Container Type

To the best of our knowledge, this is the first study of food weight and calorie estimation that considers food served in bowls. In our proposed system, we focused on food served in bowl shape containers, but the methods explained can be applied to plate containers as well as other types of containers. Other existing works [15] required the use of plate container with a specific color. For example, [22] assumed that the food is served in a white plate and the food items in the plate should be separated. With respect to their works and results, in this study, we tried to explore a different aspect with foods served in bowls, from a single image, without the use of calibration items.
From the experiments, we found that our weight estimation has some limitations coming from the camera angle (angle of view and angle of shoot). When the food image is not properly taken from the top, the system fails to detect the container shape and the chopsticks in the image (Fig 13).
To solve this problem, we will try to get the camera angle by either extracting more data from the EXIF metadata, or get the angle from the smartphone accelerometer sensor.

3.4.2 Food Serving Style and Containers

In this work, we assumed that foods are served to fill the containers. However, in real-life, foods are rarely served to fill the entire volume of the containers. There is almost always a space between the open-top of the container and the foods. This fact is probably the cause of over-estimation observed during the experiments. Another issue of the proposed system appears in the situation where a food menu is composed of many dishes or having one container for each food item of the menu. In such a scenario, the system is limited and cannot estimate the weight and calorie of all the dishes by using a single image. In this case, our target is



(a) Original image.

(b) Edges detection.

(c) Thresholded image.

(d) Output image.

Figure 13: Processing steps for container diameter and chopsticks length measurement.

only the main dish served in the largest bowl container. To estimate the calorie of the entire "set-menu", the system needs to get the image of each dish of the menu and process them separately. However, this approach might be tedious for a menu with more than two dishes.

3.4.3 Edge Detection Limitation

The shape detection steps can sometimes fail to detect the edge of the containers and the chopsticks. These detection failures can be caused by various environmental parameters such as the lighting conditions, illuminations and the color of the eating tables or containers [31]. In our work, all the photos were taken by the participants in their usual and natural eating environments, without any environmental bias. The use of such images taken in these non-pre-defined eating

environments, with no control on the illuminations, can affect the result of the shape detection mechanism. However, despite the use of these food photos taken without any environmental bias, the proposed system achieved a good edge detection rate of approximately 64%. Indeed, out of the 79 properly taken photos (described in Section 3.3.2), the system succeeded to correctly detect the edges in 50 images used for the evaluation of the overall system. We believe that this rate of edge detection without environment bias is acceptable to achieve an effective food journaling system.

3.4.4 Food Type Detection

In our current work, we did not implement any food type recognition techniques. We assumed that the food type is entered manually into the system. The motivation for this approach is that we firstly focused on the most challenging part of the food calorie estimation systems, which is the volume and weight measurement from food images. As mentioned in Section 3.2.3, there are already plenty of studies that use computer vision techniques to recognize food type. However, few studies proposed a method to compute food weight from images.
In recent years, there is also a growing interest on the use of machine learning algorithms to identify food type. For example, Pouladzadeh et al. proposed a system that uses deep convolutional neural networks to classify 10000 high-resolution food images for system training. Their results show that the accuracy of this method for food recognition of single food portions is about 99%. Although food recognition is an important step in the process of calorie estimation, in this work, we prioritized to propose an approach to overcome the most challenging step of this process. One of our plans for future work is to combine a deep learning classification algorithm with our current weight estimation system, in order to realize a fully automated food calorie estimation system.

3.5 Chapter Summary

In this chapter, we presented our food weight measurement and calorie estimation system based on image processing, using the smartphone camera and chopsticks as measurement reference. The system, also, constructs a food journal to keep track of daily consumed meals. We exploit image-processing techniques and the EXIF metadata of the food image (camera focal length, sensor size) to measure the food container size, determine the food volume, get the food weight and estimate the calorie by using density and nutrient database information of that particular food. An important aspect of this process is the use of chopsticks, which suppress the obligation of carrying and using calibration objects as shown in many existing systems. The conducted experiments show tenable results from the system which achieved an average relative error rate of 6.65% for the weight measurement, and 6.70% relative error rate for the calorie estimation. We believe that the proposed method could be used as helping tool for use in treatment of obesity, overweight and diet-related disease.

4. GIFT: Glove for Indoor Fitness Tracking

The previous chapter described our NIES system for food weight and calorie assessment. In this new chapter, we will present GIFT, our system for tracking and assessing physical activity.

4.1 Introduction

The last decades have seen a growing interest in systems that encourage and support people to the regular practice of physical activity in general, and fitness activity in particular. With the recent development of wearable computing and sensors technology, a wide range of ubiquitous systems have been developed by researchers and commercial industries to support and motivate people into 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 [55, 56]. However, the majority of these existing ubiquitous systems are either limited to only track aerobic exercise (biking, running) [17, 18], or focus only on muscle and strength training (push-up, chest press) [19, 20]. 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 tend to limit or constraint the users to perform only within the predefined set of exercises. We, therefore, propose to develop a system that would take away these constraints from the fitness practitioners and help them achieve their fitness goals.
Our proposed system is motivated by the following two observations:

- 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 pressure, protecting the hands from calluses and blisters, or increasing lift power [57].

- During workout, for most fitness activity (except for aerobics activities like running), there is always one or multiple interactions between the gloves

Figure 14: Examples of commonly performed fitness exercises.

(hands) and the athlete's body or between the gloves and the workout materials.

With that in mind, 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 14 shows some examples of commonly practiced fitness exercises with the red dots representing the area of interactions between the hand palm and the environment.

Our proposed smart-glove (Figure 15) based system 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 [39, 58], 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 (Fig 16 - left side) will produce a pressure distribution different from the distribution of a cross-body push-up (Fig 16 - center) or a reverse hand push-up (Fig 16 - right side). This kind of distinction among the same type 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:

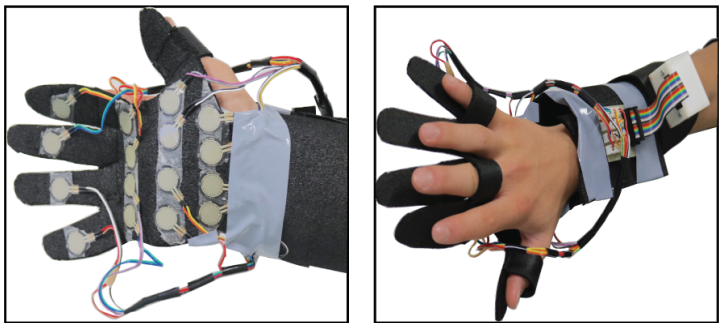


Figure 15: Inside and outside views of the smart-glove prototype.

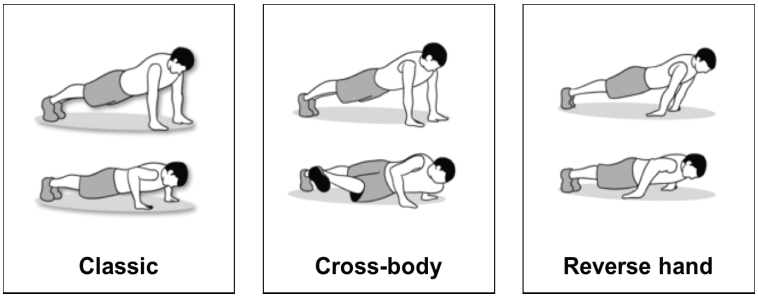


Figure 16: Three distinctive types of push-ups.

- Recognizes the activity being performed,
- Counts the number of repetitions (reps),
- Estimates the calorie burned out by each exercise set,
- Recommends future exercises to achieve users' goal.

In this dissertation, we focus on the design, the activity recognition, and the exercise counting performance of our smart-glove system.
The core contributions of this work are summarized as follows:

- *Novel wearable sensing approach*: We introduce a new resistive pressure glove that uses FSR sensors suitably mounted in the inside part of a sport glove to read real-time bio-mechanical information from the hand palm to recognize and count fitness activity.

- *Single activity detection*: By analyzing the pressure distribution applied over all the palm surface during a workout session, we can recognize most exercises of the following 4 different groups of fitness training:
  - flexibility training (e.g., side lunge stretch),
  - dynamic strength training (e.g., squat),
  - static strength training (e.g., plank),
  - 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 Chapter 4.3. Using the Ensemble subspace KNN 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%.

- *Exercise counting approach*: For each exercise set, by investigating the signal from the glove's sensors, the system automatically counts the number of exercise repetition 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%.

4.2 Design and architecture of GIFT

This section explains the design and architecture of our smart-glove system.
The GIFT 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:

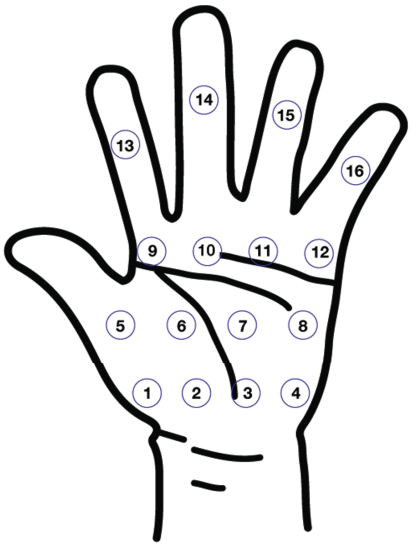


Figure 17: The sixteen locations on the glove where the force sensitive sensors are positioned to read the pressure distributions.

- *Sensing zone*: A smart-glove that well covers 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.
- *Portability*: The device should work in any fitness center or workout environment. Also, it should be lightweight and easy to wear.
- *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.

4.2.1 The force-sensitive fitness glove

The smart-glove contains a set of 16 force sensitive resistors (FSR) sensors which are mounted on the palm of the glove as depicted in Figure 17. The 16 FSR

sensors are used to read the forces applied to the palm when a user is performing a fitness exercise. FSR sensors are commonly used to detect physical pressure, squeezing and weight. They are easy to use and low-cost. In this work, we utilized the Interlink 402 Short Tail model [59]. Each sensor has an active sensing area of 12.70 mm diameter, 0.46 mm thickness, and can sense force 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 15).
As 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 distributions 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.

4.2.2 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 $2k\Omega$, and a small 400mA/h Li-Po battery.
The Adafruit Feather 32u4 Bluefruit is an Arduino-compatible + Bluetooth Low Energy (BLE) with a built-in USB and battery charging module [60]. It has a 10 bit resolution of AD converter and can read raw digital value from 0 to 1023. The DSU samples and transfers sensing data via BLE at 5 Hz. The operating supply voltage is 3.3V and it is powered by a small 400mA/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 [61].
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

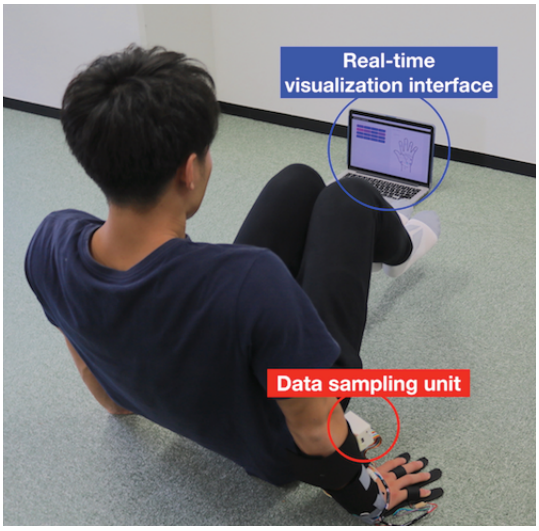


Figure 18: A user performing a knee-pull-in exercise. The real-time visualization laptop and the DSU are indicated by the blue and red circles respectively.

his/her training session. This hardware portion weighs approximately 101 g (16 sensors + wires $\approx$ 43 g and the DSU with 3D case + battery $\approx$ 58 g.)

4.2.3 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 heatmap-like color distribution on the glove image on the display. Figure 18 shows an example of a user doing a knee-pull-in exercise while watching at the visualization UI tool (Fig. 19).

4.3 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

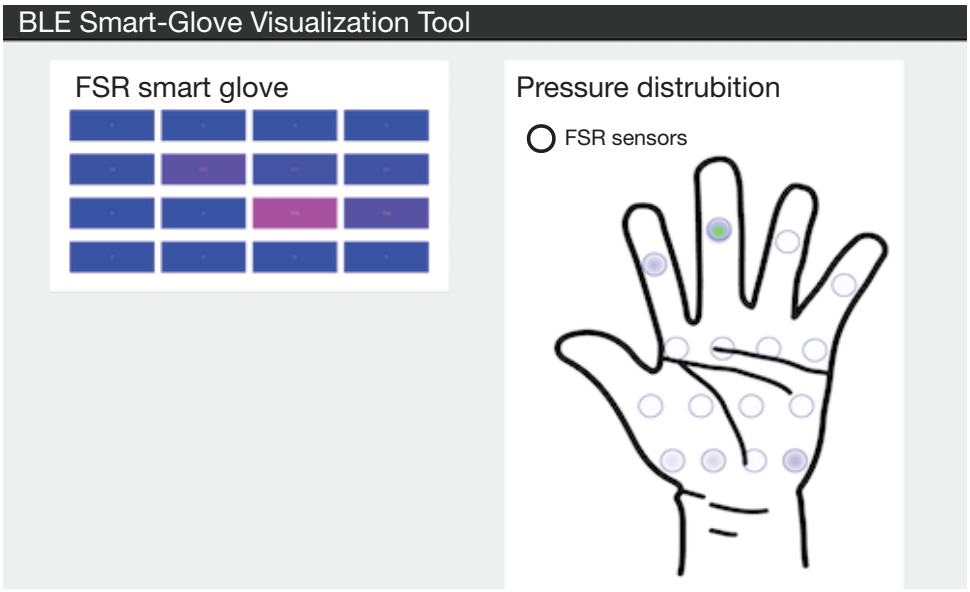


Figure 19: Screenshot of the visualization UI tool for an exercise.

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:

- Exercises with less risk of injuries: Since our participants are graduate students with their own academic schedule, we prioritized exercises with minimum injuries' risk.

- Exercise group that target full body muscles training: Most fitness programs provide workout plans that focus on full-body basic set of muscles which are biceps, triceps, chest, legs and abs [**?**]. Therefore, we chose to focus on common exercises suitable for both amateurs and professionals that workout these full-body muscles.

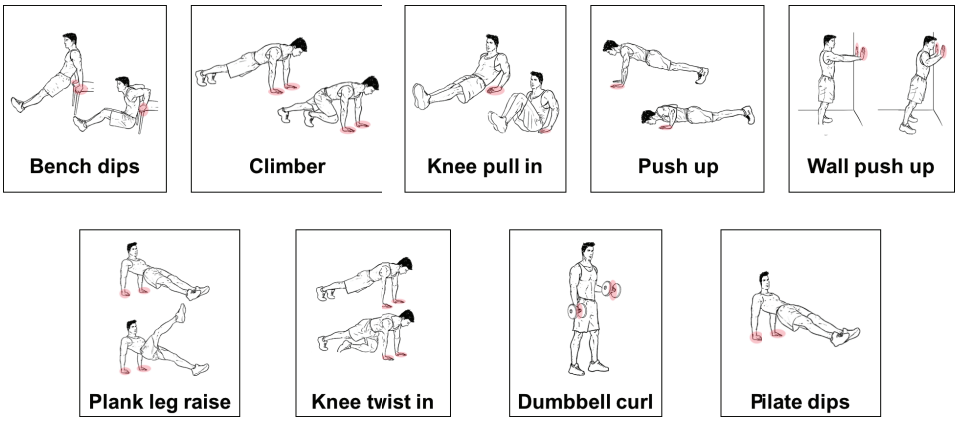


Figure 20: Exercises performed during the experiments, except the side-to-side lunge. Red areas represent the interaction surface between the glove and the workout environment.

4.3.1 Participants 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 to 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 to 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 smartphone fitness application, and they were highly motivated to see how our smart-glove could help them to improve their workout performance [62].

Table 8: 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

4.3.2 Experiment setup and tasks

The participants were asked to perform the 10 selected fitness exercises while wearing the 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 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 wants to perform the exercise. Table 8 lists the 10 exercises along with their corresponding target muscle group and Figure 20 shows a sketch of each exercise with the red area representing the interaction zone between the glove and the environment. Depending on each participant pace, the overall experiment lasts for 1-hour to 1-hour-and-half.

4.3.3 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 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 stars' 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.

4.4 Fitness Activity recognition: Methods and Results

The raw data from the smart-glove is a stream of time-series data of the 16-channels 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.

4.4.1 Features 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-channels 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 Equation (10):

$$weight(t) = \frac{\sum_{i=1}^{16} channel_i}{16} \tag{10}$$

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 domains features:

- Mean of each channel
- Standard Deviation for each channel
- Number of above mean crossing of $W(t)$
- Number of below mean crossing of $W(t)$
- Number of peaks of $W(t)$
- Skewness of $W(t)$
- Kurtosis of $W(t)$
- Band power of $W(t)$
- Mean frequency of $W(t)$
- 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 [63, 64]. For example, the mean crossing feature has been heavily used in human speech recognition and handwriting recognition problems [63]. Statistical features such as the mean or kurtosis are frequently used because of their simplicity and high performance across several human activity recognition problems [65].

4.4.2 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 KNN method, which achieved the best recognition results. It is well-known that Ensemble methods can be used to improve better predictive performance that could be obtained from any of the constituent learning algorithms alone [66].
To assess the robustness of the classifier against known and unknown users, we performed the following two evaluations.

"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 $F1$ score is presented in Figure 21a. 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 F1 score was 0.889. It is worth pointing out that only three exercises (pilate dip, plank leg, and climber) have their recognition rate lower than 80%.

"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 $F1$ score are shown in Figure 21b.
The person independent result shows an overall accuracy of 82.0%, with a F1 score of 0.830. This suggests that our system works well across different users.

In this case, also, climber and plank leg have the lowest accuracy rates of 58.3% and 60% respectively. Five exercises have their recognition rate lower than 80% (pilate dip, knee twist, side lunge, plank leg, and climber).

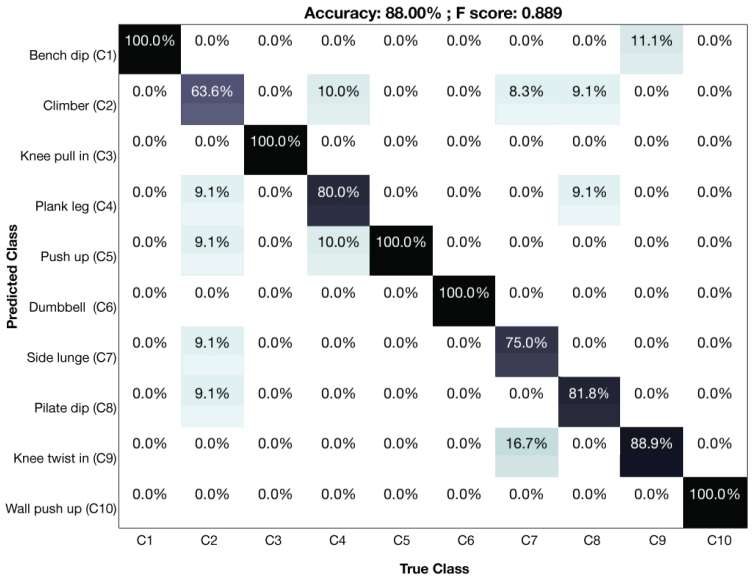
For both evaluations (leave-one-session-out and leave-one-participant-out), the overall recognition results indicate highly acceptable levels of exercise recognition accuracy which are higher than 80%. Most of the misclassification happens with exercises such as climber and plank leg raise. We also note 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 style, hand size, etc. We believe the system could be more robust against new users if we provided with 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.

4.5 Counting repetitions: Methods and Results

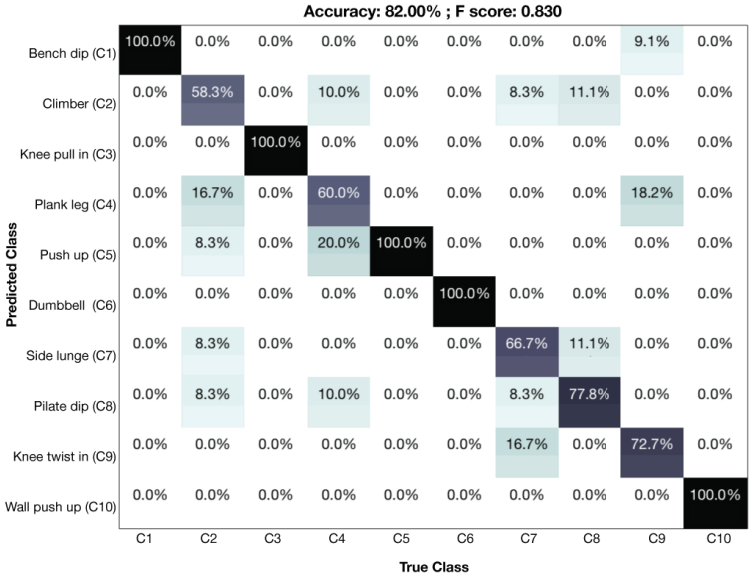
The goal of our proposed smart-glove system is to count how many repetitions a user has performed during an exercise set. To fulfill this goal, we designed a counting algorithm that uses the raw data from the smart-glove to count the number of exercise repetitions during fitness sessions. An exercise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

- Step 1: conditioning of the signal $W(t)$
- Step 2: peaks detection and elimination
- Step 3: repetition pattern detection and similarity match.

Figure 22 depicts the different steps of the counting algorithm. The dashed box indicates the activity type (exercise name) obtained from the activity recognition



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



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

Figure 21: Average recognition result for (a) person dependent and (b) person independent evaluation.

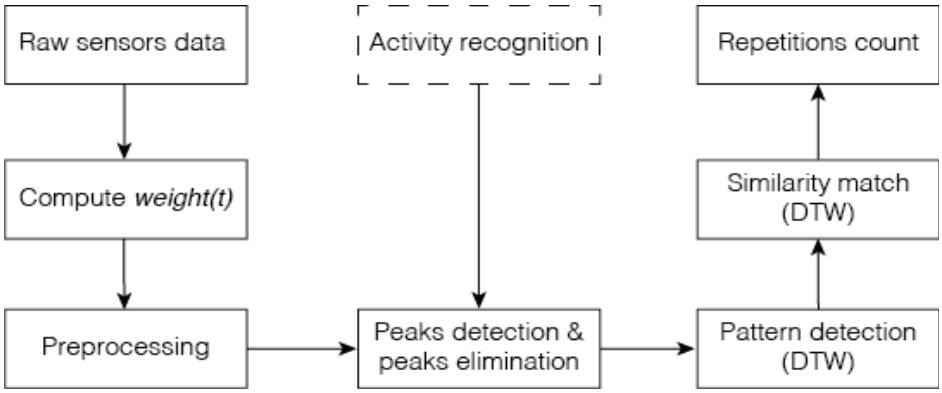


Figure 22: overview of the DTW-based repetition count algorithm.

step described in the previous Section 4.4. The counting algorithm notations along with their descriptions are summarized in Table 9.

4.5.1 Weight signal conditioning

The signal $W(t)$, as explained in the previous Section 4.4.1, is the mean of the 16 channels computed over each row of the time series data. $W(t)$ is computed using Equation (10), and normalized according to its mean value and standard deviation.

For each exercise set, the acquired $W(t)$ signal is smoothed with an eleven points fourth order Savitzky-Golay (SG) smoothing filter [67]. 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) [58]. Finally, the smoothed signal is passed to the next step (Step 2), for peaks detection and elimination.

Table 9: Algorithm notations

Notation	***Description***
$W(t)$	The mean of the 16 sensors for each time sample.
$\Delta period$	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.
$distanceDTW_{pat}$	Smallest normalized distance between matching RC.
$thresh\_dtw$	Threshold value used to accept or deny matching.

4.5.2 Peaks Detection and Elimination

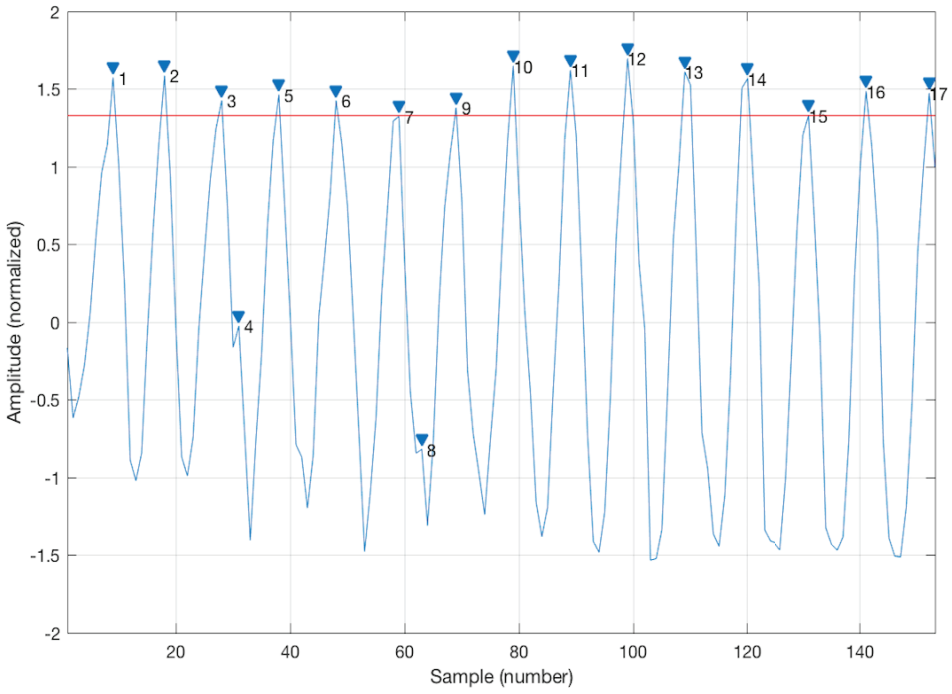
In this step, we exploit the local maxima (peaks) of the signal $W(t)$ to find repetition candidates and to reject peaks that are significantly low to be candidate peaks. The intuition behind our approach is that most of the exercise repetitions result in strong peaks.

Figure 23a 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 fitness exercise signal is quite easy (the numbers and the blue triangles represent all the candidates 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 23b).

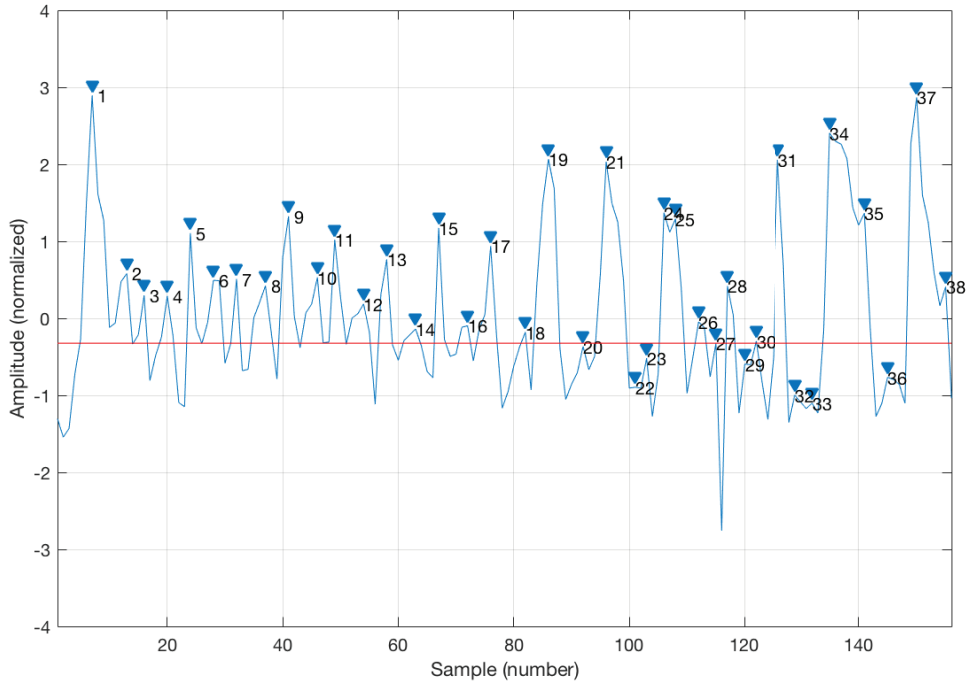
Therefore, our algorithm has to be efficient and global to handle any exercise repetition signal from the smart-glove. To that end, in this step, we need to find and reject peaks that are not generated by actual repetitions such as peaks caused by fatigues and sub-repetitions.

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 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 signals processing problems [68]. It appears that the percentile method is more efficient than setting absolute threshold values, as demonstrated in [19]. The reason is that setting a fix lower threshold value will lead to a selection of many candidate peaks, but might also contain multiple false candidate peaks; while a fix higher threshold will only detect high amplitude peaks repetitions. Therefore, applying an appropriate n-th percentile technique appeared to be a good technique for a 1st level peaks filtering. In our study, generally peaks corresponding to fatigues or sub-repetitions fall below the $20th$ percentile peak. In Figures 23a and 23b, 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 $\Delta period$ seconds away from the previously-accepted, 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 $0.4sec$), one of them is not an actual repetition peak.
The value of $\Delta period$ (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 $\Delta period$ is equaled to $1.0sec$, and for push up exercises it is equaled to $2.0sec$.
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.



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



(b) Example signal with inconstant peaks amplitude and irregular shapes.

Figure 23: Examples of repetition signals.

4.5.3 Repetition Pattern Detection and Similarity Match

At the end of the two previously described steps, we already have a naive repetitions 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 examine the signal $W(t)$ to find out how many time 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 proved to be effective for speech recognition [69] and has been intensively used in previous studies related to activity recognition [65], human motion [70] and bio-informatics [71].
In our counting algorithm, we utilize DTW to:

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

Compared to some previous studies as in [21], 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.

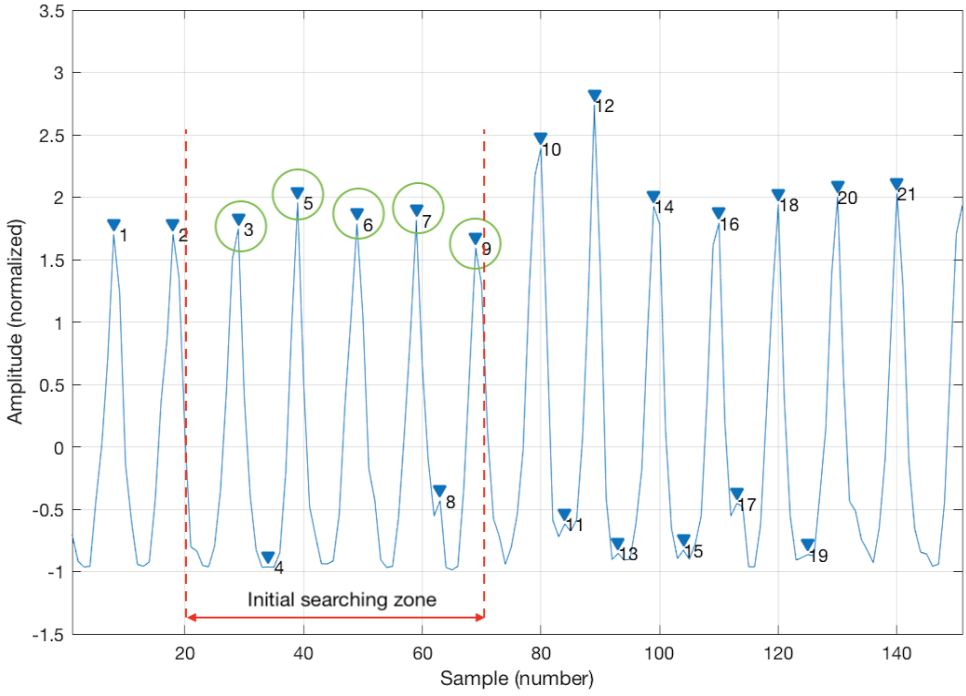
Find the repetition pattern (Phase 1)
In this phase, we empirically defined a time window of length 10 *seconds* starting from second 4 to second 14 of $W(t)$. We called this time window the searching zone (SZ). The algorithm will look for the repetition pattern only inside the SZ. The reason behind 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 hands' 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 9). 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, 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 matching among the 3 repetitions candidates, then the algorithm will do the following:

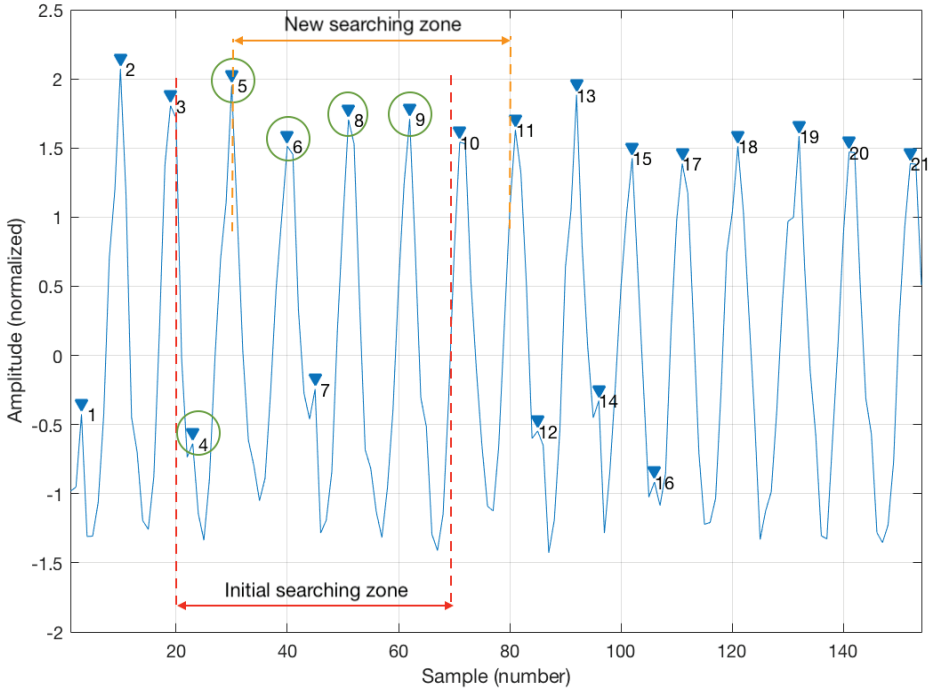
1. reject the candidate peak CP_3;
2. define a new searching zone SZ of 10 seconds starting from the candidate peak CP_4;
3. repeat the pattern searching process in the new SZ.

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 similarity match.
Figure 24a shows an example of a pattern repetition found in the initial searching zone (red color) and Figure 24b is another example where the pattern repetition is found in the self-generated searching zone (orange color).

Similarity matching and repetition count (Phase 2)
In this phase, we utilize the DTW algorithm to compare each repetition candidate



(a) Repetition pattern found in the initial searching zone.



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

Figure 24: 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.

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 matching. The higher the value of β, the more divergent (or dissimilar) sequences are considered as matching and the algorithm over-performed. Therefore, a reasonable value for β has to be determined for optimum performance. Through a try 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 Equation (11):

$$thresh\_dtw = \beta * distanceDTW_{pat} \tag{11}$$

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 (Equation (12)).

$$RC_j = \begin{cases} accept & \text{if } dtw(RC_j, R_{pat}) \leq thresh\_dtw \\ reject & \text{if } dtw(RC_j, R_{pat}) > thresh\_dtw \end{cases} \tag{12}$$

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

4.5.4 Counting algorithm evaluation and results

To evaluate the efficiency of our repetition counting algorithm, we utilized the set of data collected during the experiment described in Section 4.3. We also used the videos recorded during this experiment as Ground truth data (GT). For each exercise set, we used the algorithm to compute the number of repetitions for each participant and compared these number with the actual number of repetitions obtained from the recorded videos. Out of the 10 exercises, one exercise (pilate arms) was a non-counting exercise. Therefore, we evaluated the counting algorithm over 9 fitness exercises. Indeed, during the pilate arms 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:

- video of $P3$ performing the knee twist in exercise;
- video of $P4$ performing the plank leg raise exercise;
- video of $P7$ performing the climber exercise;
- 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.

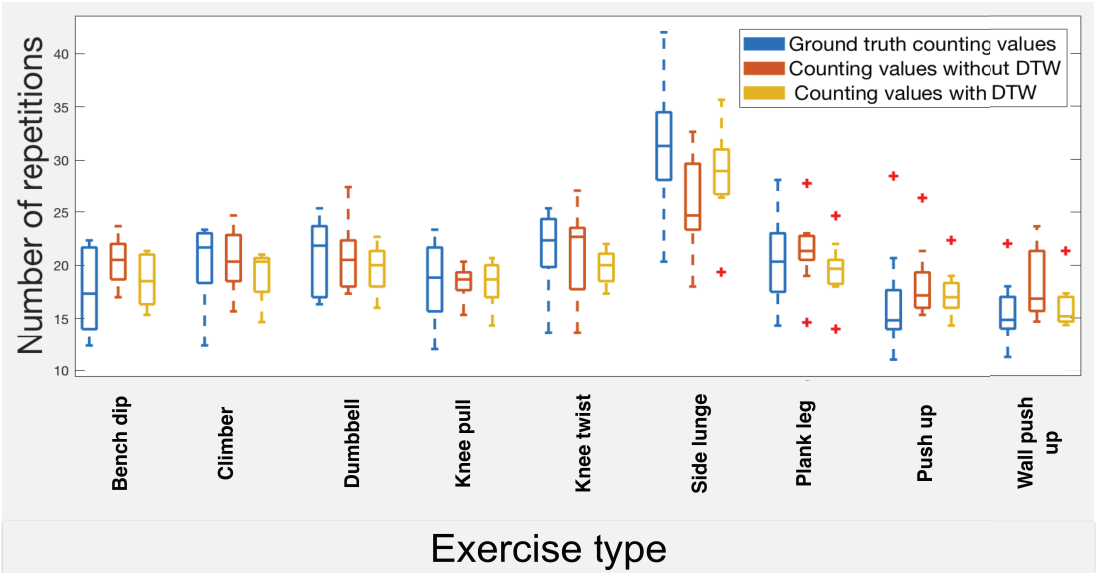


Figure 25: Average exercise repetitions count from the ground truth and count without and with the DTW.

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 over-counted the number of repetitions.

The box plot in Figure 25 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 under-counted. Furthermore, for each participant we computed the average repetitions count per exercise (remember that each 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(13). The parameter mis_count allows us to rapidly understand the performance of the algorithm after each step (Step 2 and Step 3).

$$mis\_count = GT - Alg \tag{13}$$

Table 10 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 over-counting of $+5$ up to $+17$ the actual number of repetitions (e.g., dumbbell of $P3$; side lunge of $P7$). We also notice some high under-counting, especially with the knee-twist-in exercise with -10 of the exact repetitions number, in the worst case. We defined as a "high over-counting" any repetitions count number obtained from the algorithm that is higher than 4+ the number of actual repetition. The same definition is applied to the undercount cases, where the "high under-counting" starts from -4 of the number of actual repetitions. These high over-counting and high under-counting results attest that after Step 2, the algorithm is still sensitive to timing and speed variations. The high over-counting and under-counting repetitions numbers are highlighted in red and blue color respectively in Table 10.

Although the peaks count and time-elapsed 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.

Table 10: 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

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 are almost equal to the number of repetitions observed in the ground truth data. The box plot in Figure 25 depicts the average number of repetition from the ground truth (blue color) and the algorithm after integrating DTW (orange color).

Table 10 also provides for each participant the result of the *mis_count* per exercise after Step 3. The highest over-count was about 6.3+ the actual repetitions 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.

Overall results

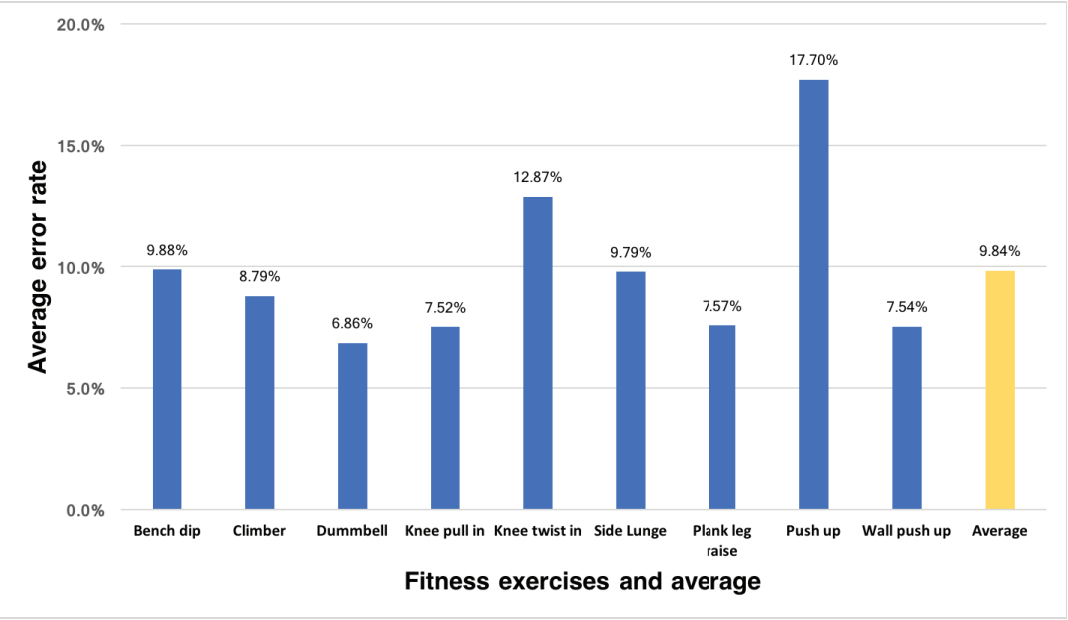
In evaluating our algorithm, we computed another parameter: *error rate* in percent.

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 repetitions count obtained after integrating the DTW (Step 3). Figure 26a plots the average counting error rate of each activity, while Figure 26b shows the average counting error rate of each participant.

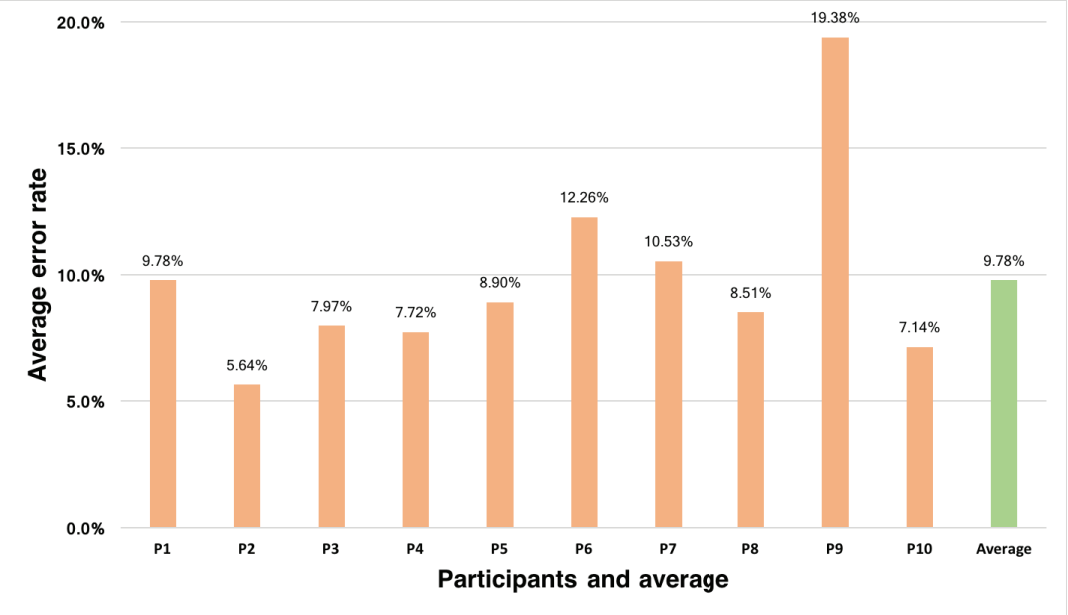
Independently of the participant and exercise type, there is no result of an error rate higher than 20%. With most participants, the push-up exercise has the highest error rate of 17.70%. This is explicable since the push-up exercise is the most exhausting exercise to be performed over 30 *sec*. Some participants pause and resume while in the middle of the 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 during 30 *sec*. Two other participants performed the push-up in a different style, which consists of making sub-repetition movement after each normal repetition. This style (a.k.a "military push-up") is generally performed by non-amateurs 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 month 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.



(a) Average error rate for each exercise.



(b) Average exercise count from the ground truth and count after Step 3 of the algorithm.

Figure 26: Average error rate by exercise and by participant.

4.6 Discussion

4.6.1 Users 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 comfortable. Figure 27 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-stars score and the other rated it to 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 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 these feedbacks, 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.

4.6.2 Number of FSR

As mentioned in Section 4.2, 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 at reducing 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. 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

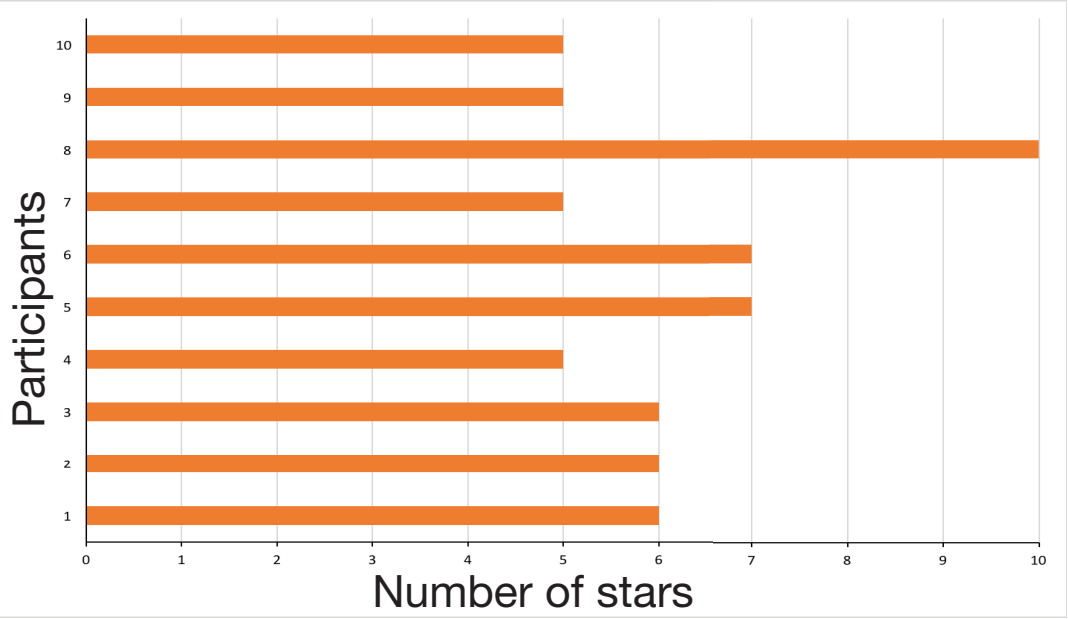


Figure 27: Score of the 10-stars scale comfort level for each participant.

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 11 shows the average force values applied to each sensor per exercise, and Figure 28 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 11).
If for a given exercise, the average value of a sensor is:

- higher than or equal to M_{TF}, we consider the sensor as "always" activated during the exercise (green circle in Figure 28).
- lower than M_{TF} and higher than $M_{TF}/10$, then the sensor is considered as "sometimes" activated (orange triangle).
- 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

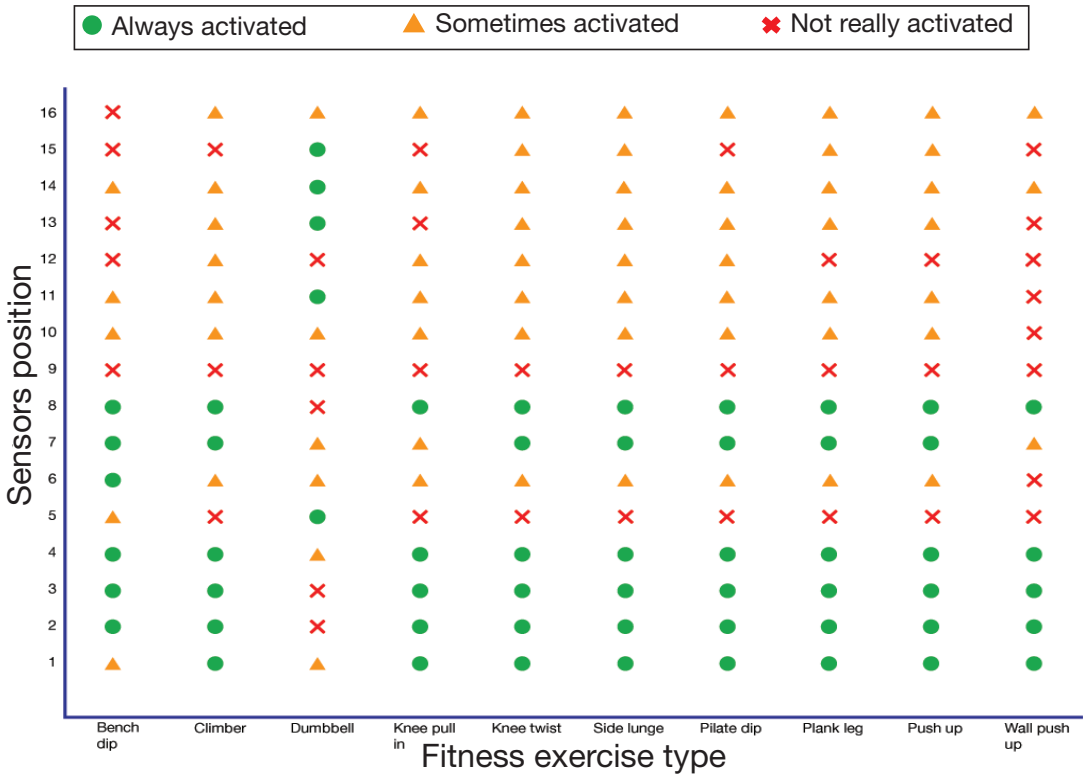


Figure 28: Sensors activation status during fitness exercise.

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 sensors values are negligible for most exercises. The sensors on the fingers (position 13, 14, 15 and 16) can also be removed or slightly moved down because they are frequently inactivated or few times activated 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.

4.6.3 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,

Table 11: 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

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'll be able to track pressure-less exercises, as well as improving the exercises' recognition and counting performance of the proposed system.

4.6.4 The ultimate goal

As for now, we have not yet reached our ultimate goal of providing the amount of calorie burned during an exercise. However, this last step can quickly be achieved by using the obtained information: the exercise type and the number of repetition.
Indeed, many online databases of thousand exercises such as FitClick [72] provide calorie information of workout exercises assuming that the exercise name, the number of repetition, and the exercise duration are already known. Thereby, our immediate future work will consist to connect our system to those databases and extract the calorie expenditure information.
Regarding calorie burned from other daily physical activities, our plan is to combine the GIFT system to existing estimation systems that target those activities.

4.7 Chapter Summary

In this chapter, 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 commons 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 hands palms 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.

5. Conclusion and Future Work

5.1 Summary

The goal of our work was to propose and implement ubiquitous systems to support people in the prevention and treatment of overweight and obesity. To ensure the usefulness of the systems we wanted to realize, we initially focused on the main recommendations of the World Health Organization (WHO) to prevent overweight and obesity: (1) control and limit calorie food intake, and (2) engage in and track regular physical activity. For these two recommendations, we have designed and implemented two ubiquitous-based systems which are easy-to-use, easy-to-carry, suitable for daily usage and designed with commodity devices.

For the first WHO's recommendation, we developed an image-based system that we called NIES (Nutrient Intake Estimation System). NIES estimates the food weight and calorie of a food using the picture of that food taken with a smartphone camera. The novelty of this system is that it used chopsticks as a measurement reference to determine food size portion. Since the eating tools are already available in the most eating environment, the users of the system are not required to always carry calibration cards or other measurements objects. Our experimental trial for the NIES ran on 50 different food pictures has shown highly acceptable results. We achieved an average error rate of 6.65% for the food weight measurement, and 6.70% for the calorie estimation.

For the second WHO's recommendation, we designed and implemented a smart-glove based system, called GIFT (Glove for Indoor Fitness Tracking), that monitors and tracks physical sports activity. The GIFT system reads, records, and analyses the time series pressure distribution signal produced by the direct interaction between the users' hand palms and the workout environment. We used these time series data to assess the users' workout performances. We originally designed the smart-glove with 16- force sensitive resistors (FSR) to ensure a full coverage of the sensing zone. However, the results of this study showed that the number of FSR can be reduced to 12 or 10. Our validation experiment on the GIFT system performed with 10 participants over 10 ordinary fitness exercises showed an average recognition rate of 82.00% and an average repetition count error rate of 9.85%.

Overall, the two systems proposed in this study have the following core contributions. Firstly, we demonstrated that eating tools, which are found everywhere, can also be used as a measurement reference instead of the usually carried calibration cards. Secondly, while most existing studies focused on non-mixed food served on plate containers, we showed that mix food served in bowl shape containers can also be measured by utilizing the image of that food. Thirdly, we presented a new wearable sensing device that uses bio-mechanical information from the hand palm to monitor, track and evaluate fitness session.

5.2 Future work

For future work, we plan to improve the food weight estimation method by integrating the distortion coefficient of the user personal smartphone. Also, we intend to make the NIES system image processing algorithm more robust in order to recognize other eating tools such as spoon, fork or knife. Furthermore, for the GIFT system, we'll investigate several engineering approaches to reduce the number of FSR needed, minimize the electronic footprints, and maximize the coverage of the hand palm.

Acknowledgements

Upon the completion of this thesis, I would like to express my deepest gratitude to all those people who have made this dissertation possible.

First and foremost, I would like to thank my advisor Professor Keiichi Yasumoto for giving me the opportunity of joining his research lab and working under his supervision since 2013. During my years with him at the Ubiquitous Computing Systems Laboratory (Ubi Lab), he taught me a lot about research and team work and I especially thank him for his help on my writings and for carefully reading and commenting on countless revisions of all my manuscripts. His sincere but kind personality makes him the most respectful professor I have ever known. I also thank him not only for his consistent encouragement, guidance, and trust. His diverse knowledge and insight has helped me to outgrow my limits.

I would also like to thank Associate Professor Yutaka Arakawa, my advisor, who has an endless enthusiasm and understands me the most. He provided me a valuable lesson on how to work with people inside and outside my area and how to nourish and exchange ideas from different fields to obtain a better success. His encouragement pushed me to always surpass my limits. Arakawa Sensei, I would like you to know that I profoundly appreciate every single thing that you have done for me.

I also thank Assistant Professor Hirohiko Suwa. His valuable suggestions, questions, and support helped me a lot to improve my research. Thanks to Assistant Professor Fujimoto Manato and Assistant Professor Teruhiro Mizumoto for their daily encouragement and friendly talks. I also wish to express my gratitude to Prof. Yoshinobu Sato for reviewing this dissertation and providing feedback for improvement.

I thank all of the Ubi lab members and staffs who benefit me with their kindness, encouragement, and friendship. These five years with you, in this warm laboratory, will remain forever engraved in my mind.

Special Thanks to Assistant Prof. Doudou Fall, and PhD. Babatunde Ojetunde. They are the big brothers from mother Africa, who always supported me in many ways including my research.

Thank from the bottom of my heart, to my two buddies William-Fabrice Brou and Cedric N'djabli Konan. You were and are still there whenever, wherever I needed you. Thanks brothers.

To my parents and my lovely wife Marie-Gertrude Epse Akpa who never leave me alone. Whenever I have any problem or feeling down, they are always there to cheer me up. Out of sight, but in my heart. I love you all.

References

[1] World Health Organization. Integrated chronic disease prevention and control, 2018.

[2] EASO. Obesity facts and figures: World health organization, 2016.

[3] J Blundell and Neil A King. Exercise, appetite control, and energy balance. *Nutrition*, Vol. 16, No. 7-8, p. 519, Elsevier, 2000.

[4] Amy F Subar, Laurence S Freedman, Janet A Tooze, Sharon I Kirkpatrick, Carol Boushey, Marian L Neuhouser, Frances E Thompson, Nancy Potischman, Patricia M Guenther, Valerie Tarasuk, et al. Addressing current criticism regarding the value of self-report dietary data, 2. *The Journal of nutrition*, Vol. 145, No. 12, pp. 2639–2645, Oxford University Press, 2015.

[5] Jason P Block, Suzanne K Condon, Ken Kleinman, Jewel Mullen, Stephanie Linakis, Sheryl Rifas-Shiman, and Matthew W Gillman. Consumers 'estimation of calorie content at fast food restaurants: cross sectional observational study. *BMJ*, Vol. 346, p. f2907, British Medical Journal Publishing Group, 2013.

[6] Glen B Taksler and Brian Elbel. Calorie labeling and consumer estimation of calories purchased. *International Journal of Behavioral Nutrition and Physical Activity*, Vol. 11, No. 1, p. 91, BioMed Central, 2014.

[7] Cristen L Harris and Valerie A George. Dietary restraint influences accuracies in estimating energy expenditure and energy intake among physically inactive males. *American journal of men's health*, Vol. 4, No. 1, pp. 33–40, SAGE Publications Sage CA: Los Angeles, CA, 2010.

[8] Adrian Holliday and Andrew K Blannin. Matching energy intake to expenditure of isocaloric exercise at high-and moderate-intensities. *Physiology & behavior*, Vol. 130, pp. 120–126, Elsevier, 2014.

[9] SM Willbond, MA Laviolette, K Duval, and E Doucet. Normal weight men and women overestimate exercise energy expenditure. *The Journal of sports medicine and physical fitness*, Vol. 50, No. 4, pp. 377–384, 2010.

[10] L Gemming, Y Jiang, B Swinburn, J Utter, and C Ni Mhurchu. Under-reporting remains a key limitation of self-reported dietary intake: an analysis of the 2008/09 new zealand adult nutrition survey. *European journal of clinical nutrition*, Vol. 68, No. 2, p. 259, Nature Publishing Group, 2014.

[11] The Nielsen Company. We are what we eat., 2015.

[12] Pew Research Center. Tracking for health, 2013.

[13] The Statistics Portal. Forecast wearables unit shipments worldwide from 2014 to 2022 (in millions), 2018.

[14] Yoshiyuki Kawano and Keiji Yanai. Foodcam: A real-time food recognition system on a smartphone. *Multimedia Tools and Applications*, Vol. 74, No. 14, pp. 5263–5287, Springer, 2015.

[15] Fengqing Zhu, Anand Mariappan, Carol J Boushey, Deb Kerr, Kyle D Lutes, David S Ebert, and Edward J Delp. Technology-assisted dietary assessment. In *Computational Imaging VI*, Vol. 6814, p. 681411. International Society for Optics and Photonics, 2008.

[16] Junghoon Chae, Insoo Woo, SungYe Kim, Ross Maciejewski, Fengqing Zhu, Edward J Delp, Carol J Boushey, and David S Ebert. Volume estimation using food specific shape templates in mobile image-based dietary assessment. In *Computational Imaging IX*, Vol. 7873, p. 78730K. International Society for Optics and Photonics, 2011.

[17] Mahmoud Hassan, Florian Daiber, Frederik Wiehr, Felix Kosmalla, and Antonio Krüger. Footstriker: An ems-based foot strike assistant for running. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 1, No. 1, p. 2, ACM, 2017.

[18] Mads Møller Jensen and Florian'Floyd' Mueller. Running with technology: Where are we heading? In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: the Future of Design*, pp. 527–530. ACM, 2014.

[19] Dan Morris, T Scott Saponas, Andrew Guillory, and Ilya Kelner. Recofit: using a wearable sensor to find, recognize, and count repetitive exercises. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3225–3234. ACM, 2014.

[20] Darragh F Whelan, Martin A O'Reilly, Tomás E Ward, Eamonn Delahunt, and Brian 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, Schattauer GmbH, 2017.

[21] Mathias Sundholm, Jingyuan Cheng, Bo Zhou, Akash Sethi, and Paul Lukowicz. Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*, pp. 373–382. ACM, 2014.

[22] Insoo Woo, Karl Otsmo, SungYe Kim, David S Ebert, Edward J Delp, and Carol J Boushey. Automatic portion estimation and visual refinement in mobile dietary assessment. In *Computational Imaging VIII*, Vol. 7533, p. 75330O. International Society for Optics and Photonics, 2010.

[23] Kiyoharu Aizawa, Yuto Maruyama, He Li, and Chamin Morikawa. Food balance estimation by using personal dietary tendencies in a multimedia food log. *IEEE Trans. Multimedia*, Vol. 15, No. 8, pp. 2176–2185, 2013.

[24] Keigo Kitamura, Toshihiko Yamasaki, and Kiyoharu Aizawa. Foodlog: capture, analysis and retrieval of personal food images via web. In *Proceedings of the ACM multimedia 2009 workshop on Multimedia for cooking and eating activities*, pp. 23–30. ACM, 2009.

[25] Shulin Yang, Mei Chen, Dean Pomerleau, and Rahul Sukthankar. Food recognition using statistics of pairwise local features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2249–2256. IEEE, 2010.

[26] Gregorio Villalobos, Rana Almaghrabi, Behnoosh Hariri, and Shervin Shirmohammadi. A personal assistive system for nutrient intake monitoring. In *Proceedings of the 2011 international ACM workshop on Ubiquitous meta user interfaces*, pp. 17–22. ACM, 2011.

[27] Tatsuya Miyazaki, Gamhewage C de Silva, and Kiyoharu Aizawa. Image-based calorie content estimation for dietary assessment. In *Multimedia (ISM), 2011 IEEE International Symposium on*, pp. 363–368. IEEE, 2011.

[28] Wen Wu and Jie Yang. Fast food recognition from videos of eating for calorie estimation. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pp. 1210–1213. IEEE, 2009.

[29] Joachim Dehais, Marios Anthimopoulos, Sergey Shevchik, and Stavroula Mougiakakou. Two-view 3d reconstruction for food volume estimation. *IEEE transactions on multimedia*, Vol. 19, No. 5, pp. 1090–1099, IEEE, 2017.

[30] Fanyu Kong and Jindong Tan. Dietcam: Automatic dietary assessment with mobile camera phones. *Pervasive and Mobile Computing*, Vol. 8, No. 1, pp. 147–163, Elsevier, 2012.

[31] Parisa Pouladzadeh, Shervin Shirmohammadi, and Rana Al-Maghrabi. Measuring calorie and nutrition from food image. *IEEE Transactions on Instrumentation and Measurement*, Vol. 63, No. 8, pp. 1947–1956, IEEE, 2014.

[32] Hsin-Chen Chen, Wenyan Jia, Zhaoxin Li, Yung-Nien Sun, and Mingui Sun. 3d/2d model-to-image registration for quantitative dietary assessment. In *Bioengineering Conference (NEBEC), 2012 38th Annual Northeast*, pp. 95–96. IEEE, 2012.

[33] Bo Zhou, Jingyuan Cheng, Mathias Sundholm, Attila Reiss, Wuhuang Huang, Oliver Amft, and Paul Lukowicz. Smart table surface: A novel approach to pervasive dining monitoring. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*, pp. 155–162. IEEE, 2015.

[34] Fitbit Home page. Fitbit official site for activity trackers and more, April 2018.

[35] Garmin Wearable Technology. Wearables and smartwatches garmin, April 2018.

[36] Fitbit Community. Logging strength training, Sep 2017.

[37] ActiveLinxx. Power your fitlinxx system, Dec 2016.

[38] Google Fit Help. Fit isn't tracking activities correctly, April 2018.

[39] Keng-Hao Chang, Mike Y Chen, and John Canny. Tracking free-weight exercises. In *International Conference on Ubiquitous Computing*, pp. 19–37. Springer, 2007.

[40] Qiang Ye, MirHojjat Seyedi, Zibo Cai, and Daniel TH Lai. Force-sensing glove system for measurement of hand forces during motorbike riding. *International Journal of Distributed Sensor Networks*, Vol. 11, No. 11, p. 545643, SAGE Publications Sage UK: London, England, 2015.

[41] Bo Zhou, Mathias Sundholm, Jingyuan Cheng, Heber Cruz, and Paul Lukowicz. Never skip leg day: A novel wearable approach to monitoring gym leg exercises. In *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on*, pp. 1–9. IEEE, 2016.

[42] Bo Zhou, Harald Koerger, Markus Wirth, Constantin Zwick, Christine Martindale, Heber Cruz, Bjoern Eskofier, and Paul Lukowicz. Smart soccer shoe: monitoring foot-ball interaction with shoe integrated textile pressure sensor matrix. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pp. 64–71. ACM, 2016.

[43] Ye He, Chang Xu, Nitin Khanna, Carol J Boushey, and Edward J Delp. Food image analysis: Segmentation, identification and weight estimation. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pp. 1–6. IEEE, 2013.

[44] Elder Akpa Akpro Hippocrate, Hirohiko Suwa, Yutaka Arakawa, and Keiichi Yasumoto. Food weight estimation using smartphone and cutlery. In *Proceedings of the First Workshop on IoT-enabled Healthcare and Wellness Technologies and Systems*, pp. 9–14. ACM, 2016.

[45] S.Perkins A.W. R.Fisher and E.Wolfart. Hough transform, 2003, December 2015.

[46] Emug CV. Emgu cv library documentation, December 2015.

[47] Open CV. Canny edge detector, December 2015.

[48] A. LLC. The aqua calculator, December 2015.

[49] Marc Bosch, Fengqing Zhu, Nitin Khanna, Carol J Boushey, and Edward J Delp. Combining global and local features for food identification in dietary assessment. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 1789–1792. IEEE, 2011.

[50] Ye He, Chang Xu, Nitin Khanna, Carol J Boushey, and Edward J Delp. Analysis of food images: Features and classification. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pp. 2744–2748. IEEE, 2014.

[51] Hokuto Kagaya, Kiyoharu Aizawa, and Makoto Ogawa. Food detection and recognition using convolutional neural network. In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1085–1088. ACM, 2014.

[52] Edith Talina Luhanga, Akpa Akpro Elder Hippocrate, Hirohiko Suwa, Yutaka Arakawa, and Keiichi Yasumoto. Happyinu: exploring how to use games and extrinsic rewards for consistent food tracking behavior. In *Mobile Computing and Ubiquitous Networking (ICMU), 2016 Ninth International Conference on*, pp. 1–7. IEEE, 2016.

[53] Forestry Ministry of Agriculture and Fisheries-Japan (MAFF). What is “ shokuiku ”(food education)?, November 2016.

[54] Research Service (USDA) United States Department of Agriculture Agricultural. Usda food composition databases 2016, November 2016.

[55] Kathleen F Janz, Jeffrey D Dawson, and Larry 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, LWW, 2000.

[56] Stavros Asimakopoulos, Grigorios Asimakopoulos, and Frank Spillers. Motivation and user engagement in fitness tracking: Heuristics for mobile healthcare wearables. In *Informatics*, Vol. 4, p. 5. Multidisciplinary Digital Publishing Institute, 2017.

[57] LiveStrong. The benefits of wearing weight lifting gloves, Oct 2017.

[58] Igor Pernek, Karin Anna Hummel, and Peter Kokol. Exercise repetition detection for resistance training based on smartphones. *Personal and ubiquitous computing*, Vol. 17, No. 4, pp. 771–782, Springer, 2013.

[59] Interlink Electronics. Fsr 400 series data sheet, Nov 2016.

[60] AdaFruit. Adafruit feather 32u4 bluefruit le, Nov 2016.

[61] Physical Activity Council. 2018 participation report: The physical activity council's annual study tracking sports, fitness, and recreation participation in the us, 2018.

[62] Edith Talina Luhanga, Akpa Akpro Elder Hippocrate, Hirohiko Suwa, Yutaka Arakawa, and Keiichi Yasumoto. Identifying and evaluating user requirements for smartphone group fitness applications. *IEEE Access*, Vol. 6, pp. 3256–3269, IEEE, 2018.

[63] Tâm Huynh and Bernt Schiele. Analyzing features for activity recognition. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pp. 159–163. ACM, 2005.

[64] Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*, pp. 197–205. IEEE, 2014.

[65] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, Vol. 46, No. 3, pp. 1–33, ACM, 2014.

[66] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, Vol. 33, No. 1, pp. 1–39, Springer, 2010.

[67] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, Vol. 36, No. 8, pp. 1627–1639, ACS Publications, 1964.

[68] Juha Parkka, Miikka Ermes, Panu Korpipaa, Jani Mantyjarvi, Johannes Peltola, and Ilkka Korhonen. Activity classification using realistic data from wearable sensors. *IEEE Transactions on information technology in biomedicine*, Vol. 10, No. 1, pp. 119–128, IEEE, 2006.

[69] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

[70] Jaron Blackburn and Eraldo Ribeiro. Human motion recognition using isomap and dynamic time warping. In *Human motion–understanding, modeling, capture and animation*, pp. 285–298. Springer, 2007.

[71] John Aach and George M Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, Vol. 17, No. 6, pp. 495–508, Oxford University Press, 2001.

[72] FitClick. Calories burned calculator, April 2018.

Publication List

Main Publications

1. Elder Akpa Akpro Hippocrate, Hirohiko Suwa, Yutaka Arakawa, Keiichi Yasumoto, "Food Weight Estimation using Smartphone and Cutlery", *IoT of Health '16: Proceedings of the First Workshop on IoT-enabled Healthcare and Wellness Technologies and Systems*, pages 9-14. ACM, 2016.

2. Elder Akpa Akpro Hippocrate, Yutaka Arakawa, Keiichi Yasumoto, "Calorie-Aware Food Recommendation System", *MobiSys '16 ASSET: Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion*, page 8-8. ACM, 2016.

3. Elder Akpa Akpro Hippocrate, Hirohiko Suwa, Yutaka Arakawa, Keiichi Yasumoto, "Smartphone-Based Food Weight and Calorie Estimation Method for Effective Food Journaling", *in SICE Journal of Control, Measurement, and System Integration*, Vol:10, Num:5, pp: 360-369. SICE, September 2017.

4. Elder Akpa Akpro Hippocrate, Masashi Fujiwara,Yutaka Arakawa, Hirohiko Suwa,Keiichi Yasumoto, "GIFT: Glove for Indoor Fitness Tracking System" , *PerHealth 3rd IEEE PerCom Workshop on Pervasive Health Technologies*, IEEE, 2018.

5. *(Conditionally Accepted with major revisions)* Elder Akpa Akpro Hippocrate, Hirohiko Suwa, Yutaka Arakawa, Keiichi Yasumoto, "A Smart Glove to Track Fitness Exercises by Reading Hand Palm", *in Journal of Sensors*, Hindawi, 2018.

Other Publications

1. Elder Akpa Akpro Hippocrate, Kazuki Fujisawa, Cedric Konan, Marko Trono, William Brou, Keiichi Yasumoto "CuraCopter: Automated Player Tracking and Video Curating System by using UAV for Sport Sessions", *IPSJ SIG Technical Report*, vol 2015-MBL, Num:7. ISPJ 2015.

2. Edith Talina Luhanga, Elder Akpa Akpro Hippocrate, Hirohiko Suwa, Yutaka Arakawa, Keiichi Yasumoto, "Towards Proactive Food Diaries: A Participatory Design Study", *Smarticipation2016: 1st International Workshop on Intelligent Personal Guidance of Human Behavior Utilizing Anticipatory Models*, pages 1263-1266. ACM, 2016.

3. Edith Talina Luhanga, Elder Akpa Akpro Hippocrate, Hirohiko Suwa, Yutaka Arakawa, Keiichi Yasumoto, "HappyInu: Exploring how to use games and extrinsic rewards for consistent food tracking behavior", *The 9th International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2016)*, pages 1-7. IEEE, 2016.

4. Elder Akpa Akpro Hippocrate, Edith Talina Luhanga, Takata Masashi, Ko Watanabe, Keiichi Yasumoto, "Smart Gyms Need Smart Mirrors: Design of A Smart Gym Concept through Contextual Inquiry", *UbiComp/ISWC '17 Adjunct* , pages 658-661. ACM, 2017.

5. Edith Talina Luhanga, Elder Akpa Akpro Hippocrate, Hirohiko Suwa, Yutaka Arakawa, Keiichi Yasumoto, "Identifying and Evaluating User Requirements for Smartphone Group Fitness Applications", *in IEEE Access Journal*, Vol:6, pp: 3256-3269, IEEE, January 2018.