Automatic caloric expenditure estimation with smartphone's built-in sensors

by

Nestor Stiven Cabello Wilson

A dissertation submitted to the Faculty of Melbourne Graduate School of Science in partial fulfillment of the requirements for the degree of Master in Science (Computer Science)

Supervised by:

Dr. Lars Kulik

Dr. Masud Moshtaghi

MSc. Yousef Kowsar

November 2016

Student number:

Type of project: Research Project

The total number of project's credit points: 75 points

The subject code: COMP60002 - Computer Science Research Project

Abstract

Fitness-tracking systems are technologies commonly used to enhance peoples' lifestyles. Feedback, usability, and ease of acquisition are fundamental to achieving the good physical condition goal. Users need constant motivation as a way to keep their interest in the fitness system and consequently, continue on a healthy lifestyle track.

However, although feedback is increasingly being incorporated in many fitness-tracking systems, usability and ease of acquisition are remaining shortcomings that need to be enhanced. Features such as automatic activity identification, low-energy consumption, simplicity and goals-achieved notifications provide a good user experience. Nevertheless, most of these functions require the acquisition of a relatively expensive fitness-tracking device.

Smartphones provide a partial solution by allowing users an easy access to multiple fitness applications, which reduce the need for purchasing another gadget. Nonetheless, improvements in the user experience are still necessary. In the other hand, wearables devices satisfy the usability, however, the cost of their acquisition represents an impediment to some users.

The system proposed in this research aims to handle these issues and offers a solution by combining the benefits from mobile applications such as feedback and ease of acquisition, with the usability that wearable devices provide, into a smartphone Android application. Data collected from a single user while performing a series of common daily activities namely *walking*, *jogging*, *cycling*, *climbing stairs*, and *walking downstairs*, was used to classify and provide an automatic identification of these activities with an overall accuracy of 91%, and identifying the *stairs* activities with an accuracy of 81%. Finally, the caloric expenditure, which we considered the most important metric for motivating a user to perform a physical activity, was estimated by following the oxygen consumption equations from the American College of Sports Medicine (ACSM).

Declaration

I certify that, this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text. The thesis is 17335 words in length (excluding text in images, tables, and bibliographies).

Acknowledgments

I would first like to thank to my supervisors, Professors Lars Kulik, Masud Moshtaghi and Yousef Kowsar for their guidance, motivation, insightful comments, hard questions, and for sharing their knowledge through the entire process of research and writing of this dissertation.

My sincere thanks to my parents, my brothers, and my sister for their ongoing support and continuous encouragement throughout my years of study.

Finally, I would like to express my profound gratitude to my wife Bianca for her love, unconditional support, patience and editing assistance of this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the present chapter, we start with an introduction on potential effects of technological advances in sensing systems in promoting a healthy lifestyle. Next, we enumerate the main contributions and the approach taken to fulfill the main research questions presented in this thesis.

1.1 Introduction

Despite the growing interest in fitness and healthier lifestyles, according to [1], overweight and obesity, defined as "abnormal or excessive fat accumulation that may impair health", has more than doubled since 1980. Only in Australia 63% of adults are overweight or obese [2]. Furthermore, in 2014, more than 1.9 billion adults, 18 years and older, were overweight. Of these, over 600 million were obese. Therefore, about 39% of the world's adult population were overweight, and 13% were obese in 2014.

Engaging in regular physical activities is a way to prevent overweight and obesity. However, due to modern world's barriers, people nowadays are looking for other ways to stay fit rather than the traditional aerobics and gym activities, which are now the second most popular forms of exercise [3]. Between the reasons of the decreasing popularity of aerobics and gym activities are included cost, lack of time and feeling out of place [3].

On the other hand, the number of "health and fitness" smartphone's applications (apps) is growing fast. Having increased by 62% in 2014 compared with 33% for apps in general

[4]. Moreover, smartphones which have become indispensable devices for almost every person (according to [5] the number of smartphone users in the world is expected to reach 2.6 billion by 2019) are turning into powerful machines, which no only allow for text and voice communication, but can also provide different types of information. Multiple sensors such as the accelerometer, gyroscope, barometer, and GPS are now built-in on almost every smartphone. The combination of these sensors can provide information about the activity a user is performing, and besides meaningful details about the activity per se, i.e. how fast or slow a user is walking. Nevertheless, despite the constant motivation these apps provide to the user (i.e. caloric goal reached notifications, the number of steps taken daily information), the majority of these apps are GPS-based and therefore will drain the battery at a faster rate. Furthermore, these apps require constant user interaction. Users have to select the activity they want to track, start the tracking and end it, considered as a shortcoming or something important to improve as mentioned in [6] [7].

Another kind of gadgets that are also very popular as fitness-tracking systems are wearables devices, such as smartwatches or wristbands (e.g. Fitbit HR, Samsung Gear 2, Apple Watch). As described in [8], 45 million wearable fitness bands are expected to be sold in 2017, rising to 99 million annually by 2019. In contrast to mobile apps, these devices automatically identify the activity the user is performing, but consequently, this means that a user needs to purchase an extra device in order to be able to have a good fitness-tracking system, which can be seen as a limitation for some users.

1.2 Motivation

The higher number of fitness-tracking apps and wearables does not actually reflect satisfaction from users. In [9][10] it is suggested that the early appeal and fascination with this kind of wearables does not last. As well, in [11] it is mentioned that the apps' usability may become a barrier: if people do not know how to use these apps, then apps will be inefficient, and people will give up using them.

It is also important to mention that, as suggested in [12], fitness supported by technology has evolved into a trend, becoming a lifestyle and status symbol. Therefore, individuals more likely to use this kind of technologies tend to have higher incomes and be more educated [12] [13].

Hence, one of the reasons of the present work is to extend the interest in a healthy lifestyle to anyone, indistinctly of its social-economic status, by offering a fitness-tracking system capable of providing a good user experience reachable to any pocket.

The second reason consists of to explore and exploit the current smartphone's capabilities to demonstrate the power and importance of its built-in sensors and also to point out shortcomings of these devices in the fitness-tracking area.

1.3 Contribution

The main contributions of this thesis are:

- The development and publication of an Android fitness-tracking mobile application with automatic activity recognition, low-energy consumption, and accurate caloric expenditure estimation.

- The delivery of datasets with the accelerometer, gyroscope and barometer sensors' readings from different activities such as walking, jogging, cycling, and walking up/down stairs for further analysis and research.

1.4 Approach

In Chapter 2 we give an overview of current advances on activity recogniton systems by the use of sensors. In the same way, we discuss about the gaps in the literature, and the importance of their further analysis, that consequently had led us to the following main research questions: (1) Are current smartphones technologies capable of enhancing the fitness-tracking apps usability? (2) Is it possible to accurately estimate the caloric expenditure of a person while performing an activity by the only use of built-in smartphone's sensors?

To improve the current usability provided by fitness apps, we decided to include the automatic activity recognition feature, which is a desired feature among users. Chapter 3 explains the data collection and feature generation processes. Finally we describe the classifier used and the different tests performed to find the best parameters for our activity recogntion system.

For the caloric expenditure estimation, we are using the ACSM's equations. These formulas are well-researched activity-based formulas with certain set of parameters. In Chapter 4 we perform various tests to evaluate the accuracy on the estimation of each parameter using a smartphone. Then, if the parameters of these equations can be accurately estimated, we can conclude that the final caloric expenditure estimation will be accurate as well. Chapter 4 also includes details on the implementation of a step detection algorithm based on the gyroscope sensor and the formulas used for speed, elevation, and work rate computations necessary to provide an accurate energy expenditure estimation.

The workflow of the whole system is shown in Fig. 1.1. In Chapter 5, we present discussions about the assumptions, limitations, and results from the different experiments performed in this work, and in Chapter 6, we conclude this thesis and present future works.

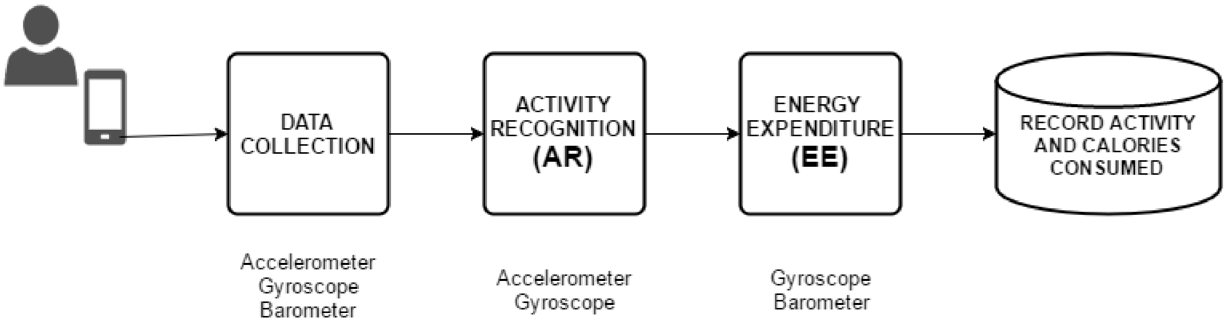


Figure 1.1: System approach overview.

Chapter 2

Literature Review

Providing a solution to battle the multiple problems and diseases that people might suffer due to unhealthy lifestyles has been the primary objective of several studies where sensors have played a meaningful role in providing the user's context necessary for the development of multiple health-related applications. Some of these applications involve looking for a relationship between chronic illnesses and environmental factors [14], identifying user's location [15] [16], monitoring physiological state [17] (heart rate, skin temperature, and respiration), and inferring physical activities. The latter application has given place to the topic of Activity Recognition Systems (ARS), which can be categorized into multiple on-body sensors and single on-body sensors ARS. In this chapter we explore the different works performed on this topic, discussing about the different approaches regarding the number and type of sensors that authors have used. Similarly, we describe the different solutions taken to develop energy efficient systems, as well as techniques to estimate energy expenditure. Besides, we explain how our work differs from prior studies and finally we indicate where future research might best be directed.

2.1 Activity Recognition Systems

ARS is a well-defined research area which has been subject of many studies. Initially, authors focused on recognizing daily activities by using multiple sensors spread across the user's body. Despite the high recognition accuracies reached by these multiple on-body sensors systems, the wearability inconvenience led to researchers to find new ways for motion identification. Therefore, a new kind of ARS emerged, which instead of using multiple sensors on different body locations, these new systems required just one body location to place a specialized device with different types of sensors inside. However, these specialized devices were later replaced for an even more convenient device, the smartphone, which due to its computational capacity and the important sensors embedded on it has been subject of different tests in recent papers of ARS. Next, we review works related to the two categories mentioned above.

2.1.1 Multiple on-body sensors ARS

The first works of this kind of systems used multiple on-body sensors, typically accelerometers [18] [19] or a combination of accelerometers with other sensors [20] [21], to classify user's movements with recognition rates up to 90%. Many of these studies have searched for the ideal number of sensors and best body locations to achieve highly accurate results. One of the most significant works about these systems [22], proposes an ARS to identify twenty daily activities such as walking, sitting, or brushing teeth, by using five bi-axial accelerometers placed on five different body locations (hip, wrist, arm, ankle, and thigh) and without researcher supervision. Their results are important because they are a step forward to create systems that could work in real world conditions, i.e. outside laboratory settings. Authors achieved an accuracy rate of 84% overall in activity recognition. However, not all the twenty activities involved reached this rate, some activities such as riding elevator up/down got results lower than 50%, and others (walking up/down stairs) achieved approximately 75% of accuracy. Therefore, the addition of other types of sensors (e.g. barometer) could lead to better results. Moreover, it was also was found that the accelerometer placed in the thigh was the most powerful at recognizing activities.

Despite yielding strong recognition rates, the high number of sensors, mostly wired [23] and spread across the user's body make these systems impractical for everyday use. Hence, the conclusions from [22] regarding the fusion of accelerometer with other sensor modalities to improve recognition accuracy, and the findings of a specific but meaningful body location, led researchers to experiment with single on-body sensors systems.

2.1.2 Single on-body sensors ARS

These systems, typically used some specialized device (e.g. Mobile Sensing Platform (MSP) or Multi-modal Sensor Board (MSB)) with seven or eight different sensors inside, such as temperature, humidity, light, accelerometer, audio and barometric pressure. However, not all the sensors necessary must be used.

The use of a specialized device carried on the waist with its accelerometer and barometer embedded-sensor enabled has allowed the identification of common daily activities such as walking, jogging, cycling, walking upstairs/downstairs and riding elevator up/down [7] [24]. Moreover, as demonstrated in [25], the use of solely the accelerometer sensor from the MSP while the user carries it on his hip, make the system able to distinguish between walking and jogging activities. In [26] they used an MSB to identify with approximately 90% accuracy eight different activities (sitting, standing, walking, walking up/down stairs, riding elevator up/down, and brushing teeth) collected from twelve different subjects. However, they reduce the number of sensors from seven to three, and results were quite accurate as well. According to their experiments, the most relevant sensors were the accelerometer, audio, and barometric pressure. In addition, they tested the system with the MSB in three different locations of the user body (wrist, waist, and shoulder). All of the locations showed to be good at recognizing the activity. Thus, this work demonstrated that it is possible to change the position of the sensing device and still achieve accurate results. However, the data was collected inside a building and not under real world conditions.

Single on-body sensors by using specialized devices reduces the inconvenience of wearing multiple on-body sensors across the body. These systems showed that with a reduced number of sensors placed in a single body location (and not necessarily always the same area) it is possible to obtain competitive recognition accuracies. However, due to the increasing computational power and addition of multiple sensors, smartphones have turned into the next generation of single on-body sensor systems for activity recognition tasks. Smartphones nowadays contain as many sensors or even more than prior specialized devices (i.e. MSP or MSB) and users carry them every day without this representing an extra burden to them.

2.2 Smartphone-based ARS

According to works mentioned above, the accelerometer is considered the most discriminative sensor for motion classification. Therefore, researchers have experimented with the smartphone's built-in accelerometer by placing the mobile device in a fixed body location: front pant's pocket to identify basic daily activities [27] or attached to the back of the user to differentiate between two types of sports (soccer vs. hockey) [28]. In contrast, other works [29] have considered position-independent systems for the movement recognition task. Daily activities have been identified with 90% accuracy overall. Nonetheless, the activities involving vertical displacements such as walking up/down stairs have been difficult to identify, being often confused with the walking activity. Hence, in order to improve the recognition of these activities, many studies have tested the fusion of the accelerometer with other sensors such as gyroscope and barometer.

The most common explored sensors' combination has been accelerometer-gyroscope [30] [31]. The gyroscope has provided and aid to the accelerometer in the activity recognition task demonstrating improvements ranging from 3.1% to 13.4.% at identifying different paces of walking, jogging, and going upstairs/downstairs activities [32]. This sensor combination has also been used to recognize hand gestures activities such as smoking, eating, and drinking coffee [33], and for other types of complex activities such as cooking, cleaning, and sweeping [34]. Nevertheless, accuracy rates for these activities were at most 50%.

With the aim of improving the accuracy of methods based on the accelerometer-gyroscope combination, several works have included other sensors to this fusion. Authors in [35] use the accelerometer, gyroscope, and magnetometer with the smartphone attached to the chest to identify fifteen activities. Despite achieving an overall accuracy above 95%, the body location used does not reflect a typical placement of the smartphone. In contrast, in [36] the same 3-sensors combination was tested with the smartphone placed in more common locations: jeans pocket, belt, arm, and even on the wrist. However, the magnetometer showed a poor role at identifying any particular movement. They also found that the accelerometer was better than the gyroscope in the activity classification for almost all locations, except for the pants pockets, where the gyroscope performed slightly better than the accelerometer.

Many more sensors have been added to the accelerometer-gyroscope combination [37] [38]. One of these additions, the audio sensor, showed a significant role in the pattern recognition job [39], leading to achieving more accurate results than by only using the accelerometer and gyroscope. Furthermore, instead of the gyroscope, other sensors have also been combined with the accelerometer for different applications. The combination accelerometer-GPS was used in [40] to reliably measure the bike's cadence at cycling. Also, the accelerometer-magnetometer fusion was used to assess the quality of training in rehabilitation exercises while the mobile phone was placed onto an NFC-augmented balance board [6].

Finally, the barometer by its own has shown to be an energy efficient solution. Besides, it has also demonstrated to be useful for detecting vertically oriented activities such as riding an elevator and climbing stairs [41]. Similarly, it was efficient at differentiating between standing, walking (on the same floor), climbing stairs, and riding elevator activities [42], and to detect if the user was idle, walking or inside a vehicle [43].

Smartphone's embedded sensors have demonstrated high performance at classifying multiple activities, which added to the pervasive feature, make to the smartphone a reliable tool to explore ways to improve user's health.

2.2.1 Smartphones as fitness-promote devices

With the aim to encourage healthier lifestyles by the only use of a smartphone, authors in [44] have allowed users to monitor their activity levels by using GSM cell signal strength to identify when the user is sitting still or walking. With the same objective, authors in [7] included a glanceable display (blooming flowers) as a visual mechanism to identify the amount of activity level achieved. Others authors gave a step forward and aggregated such activity levels over time to generate physical activity diaries [45] with the purpose of suggesting a certain exercise to the user to fulfill the amount of physical activity recommended.

Activity levels provide a rough estimation of the physical state of a user by analyzing the duration and the type of physical exercises performed during the day, week, or month. The recommended level of activity for an adult is at least 30 minutes per day five days a week for a moderate-intensity physical activity routine [46]. Despite the fact that the knowledge of activity levels measured automatically can motivate users to continue exercising, a more detailed information about the types of activities performed could increase the user's motivation.

The calories burned during an activity is an important metric as it quantifies the user's effort. This metric can be seen as a reward to the user, especially while performing strenuous exercises such as jogging or cycling. Therefore, some studies [24] have focused on the estimation of the daily calories expended in physical activities. Similarly, in [25], despite estimating the caloric expenditure, they also estimated the caloric intake (food input interface) to create a balance and provide the final amount of calories gained or consumed by the user during the day.

Great efforts have been made to promote the involvement of users in regularly physical exercises. Nonetheless, these efforts are often related to an expensive computational cost. Moreover, smartphone's short battery life continue being a shortcoming of these devices, as mentioned in [47], which needs to be recharged on a daily basis. Therefore, if we add an expensive computational application to a device with some other applications already consuming resources, memory, and battery; we will end up diminishing the user experience and consequently the system acceptance.

2.2.2 Energy efficient approaches

Despite the importance of energy efficient smartphone-based ARS, a limited number of works have implemented battery-saving solutions. Authors have considered different approaches. In [48] a lightweight classification system was developed by selecting low sampling rate sensors, inexpensive computational features, and simple machine learning algorithms. Moreover, with the aim of reducing complexity by using classifiers based on discrete variables, discretization techniques were implemented in [47]. Another approach [49] used fixed-point calculations instead of floating-point by deploying a multiclass Support Vector Machine classifier using integer parameters. By doing so, they reduced power consumption while using less memory and processor time. In [50] they leveraged the noise-resistant feature of the smartphone's gravity sensor to create a simple hierarchical ARS by avoiding expensive noise filtering processes. Lastly, in [42] they used the low-power barometer sensor to provide a highly accurate but energy efficient ARS.

2.2.3 Online activity classification

Most of the smartphone-based ARS have performed the activity identification by using standard machine learning classifiers. However, this classification has been done offline, either by downloading the sensing data from the smartphone to the computer through the USB port, or by sending the collected data wirelessly to an external server.

Only few recent works [37] [47] [48] [51] have implemented an online classification, that is to say, all the activity classification is performed in the smartphone. The development of these stand-alone mobile ARS is important due to the need of an external server is no longer required, the smartphone's battery life is preserved by not sending data continuously to an external server, and the user sensing information is less likely to suffer from external attacks.

2.3 Our proposal

Our approach aims to create an stand-alone energy efficient caloric expenditure estimator system. The smartphone placed in the user pant's pocket is constantly collecting sensing data from the accelerometer, gyroscope, and barometer embedded sensors. As most of the smartphone-based ARS presented in the literature, summarized in Table 2.1, we used the accelerometer and gyroscope sensors for the pattern identification process. However, following a similar criterion for reducing complexity than in [48], only basic time-domain features were extracted. It is important to mention that our work differs from most of the energy efficient studies previously cited, in the fact that they only identify the activity performed without considering any further application to promote fitness.

Ref.	No. Subjects	Sensors	Number of activities	Smartphone Placement	Recognition Accuracy	Calories burned?	Energy Efficient?	Online Classification?
[27]	29	Accelerometer	6	Pant's pocket	90%	No	No	No
[51]	17	Accelerometer	4	Upper arm	84.25% (Precision)	Yes	No	Yes
[50]	1	Accelerometer and Gravity	4	Pant's pocket	95%	No	Yes	No
[37]	10	Accelerometer, Gyroscope and GPS	7	Hand, pant's pocket, shirt pocket, and handbag	95% (True Positve Rate)	Yes	No	Yes
[30]	4	Acceleromter and Gyroscope	7	Upper torso and lower torso locations	88% (Precision)	No	No	No
[32]	16	Acceleromter and Gyroscope	13	Armband for jogging and pant's pocket for the rest	87.41%	No	No	No
[28]	15 (soccer) 17 (hockey)	Accelerometer	2	Back	87% (F-score)	No	No	No
[49]	30	Accelerometer and Gyroscope	6	Waist	89%	No	Yes	No
[31]	3	Accelerometer and Gyroscope	6	-	95.9% (Precision)	No	No	No
[47]	10	Accelerometer	8	Waist	98%	No	Yes	Yes
[35]	10	Accelerometer, Gyroscope, and Magnetometer	15	Chest	95.03%	No	No	No
[48]	16	Accelerometer, Gyroscope, Gravity and Magnetometer	6	Hand, pant's pocket, shirt pocket, and handbag	88%	Yes	Yes	Yes
[42]	10	Barometer	7	-	95%	No	Yes	No

Table 2.1: Smartphone-based Activity Recognition Systems.

We plan to present the daily caloric expenditure estimation while performing physical activities as our fitness-promote application. Approaches such as two regression models [52], multiple regression models [24] [53], or complex accelerometer's features [54] have been considered to achieve the same goal. However, these expensive computational techniques were implemented before considering the smartphone as a reliable activity recognition device. Thus, no consideration regarding battery-saving solutions was taken into account.

More recent papers [37] [48] [51] have introduced less complex processes to estimate the energy expenditure by using simples methods that relate type of activity with indexes such as the Metabolic Equivalent of Task (MET), Physical Activity Level (PAL) or Physical Activity Ratio (PAR) to quantify the energy expenditure. However, considering that most of the common daily physical activities consist of short time periods, the use of MET, which assumes long exercising periods, would not be appropriate as suggested by [24]. Furthermore, MET, PAR, and PAL do not capture the user's effort during strenuous activities. For example in the case of jogging, MET will provide a value to the system associated with a vigorous activity, without any consideration regarding how many slopes (which obviously will increase the activity difficulty) were on the route. Previous work [40] did capture the user's effort by employing a power measurement approach by using GPS traces and elevation data from a web service. Nonetheless, the energy expenditure estimation was only oriented to the cycling activity, and even though they could duty-cycle their smartphone's GPS receiver to save energy, the use of GPS data will not be appropriate for an energy efficient solution.

For the calories burned estimation we are using the metabolic equations from the ACSM. In [25] they follow the same approach. However, they only estimated calories burned while walking and jogging on a treadmill. Besides, the gradient and speed information required for the metabolic equations was not inferred from any embedded sensor like in our approach, but taken directly from the treadmill. Another study that also used the metabolic equations for the energy expenditure estimation and that to the best of our knowledge is the closest to ours due to they also exploited the smartphone's sensing capabilities to compute speed and grade (required for the ACSM's equations) is presented in [55]. Nonetheless, only walking and jogging activities were considered in these estimations. Hence, we believe that our work is important as if incorporates a wider range of activities into the energy expenditure estimation. Additionally, to ensure practicality of our method, we are also leveraging the smartphone embedded sensors capabilities to accomplish the activity recognition and calories burned estimation tasks, and employing energy-friendly mechanisms to increase the user experience. Finally, it is worth to mention that in contrast with the online smartphone-based ARS described in the literature, our application is available in Google Play Store. We consider this important due to the useful feedbacks people may provide regarding improvements in usability and recognition accuracies.

Chapter 3

Activity Recognition

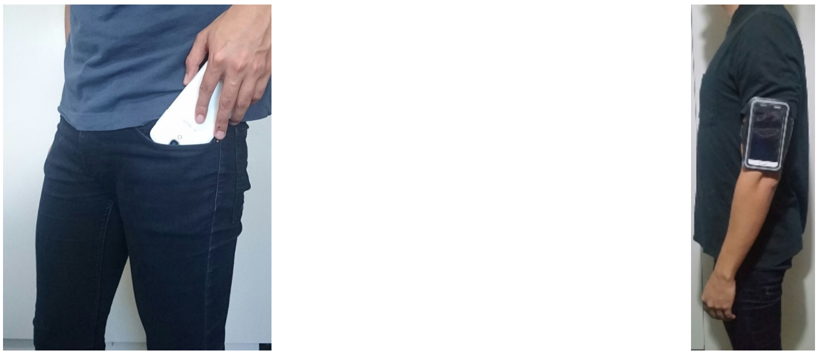
Automatic activity inference represents an important feature to address in order to enhance the user experience in fitness-related apps. Thus, in this chapter we examine the different steps to follow in order to built an activity recognition system. First, we explain how sensing data was gathered, and the necessary implementations to improve the activity labeling process. Later, we describe the explanatory variables used to construct the feature vectors that serve as input for our selected machine learning classifier. Besides, we test the performance of our classifier in two different settings and we search for the best parameters to obtain the highest accuracy result. Finally, we mention the importance of the gyroscope sensor as a complement to the accelerometer to improve the inference of some activities.

3.1 Data Collection

The smartphone used for the data collection and tests is a Samsung Galaxy S4 model SGH-M919, with Android KitKat version 4.4.4. The device is equipped with accelerometer, ambient light sensor, proximity sensor, barometer, magnetometer, thermometer, gyroscope, humidity, and gesture built-in sensors. However, the recorded data contains only information from the accelerometer, gyroscope and barometer sensors, which in prior works [27] [32] [43] have demonstrated to be useful in the activity recognition task.

Data was collected from a single participant, a healthy 30 years old male user while carrying the smartphone in his pants' front left pocket, the device was oriented with the screen facing toward the body and the micro-USB connection port facing up (see Fig. 3.1a). The participant was asked to perform his daily activities freely, but labeling a set of five activities: *walking*, *jogging*, *cycling*, *climbing stairs* (*upstairs*), and *descending stairs* (*downstairs*). Any other activity was labeled as *others.* Among the *others* activity are activities such as: *sitting*, *standing*, *riding elevator up/down*, *traveling on tram.*

In the case of *jogging* the smartphone was carried in the left arm with an armband instead of the pant's pocket, and the device was oriented with the screen facing away from the body and the micro-USB connection port facing down (see Fig. 3.1b).



(a) Position of the smartphone for the *walking*, *cycling*, *upstairs*, and *downstairs* activities

(b) Position of the smartphone for the *jogging* activity

Figure 3.1: Smartphone position during data collection.

An Android application was develepod to facilitate the data collection task (*Data collection app*). This application through a simple graphical user interface GUI (see Fig. 3.2), allows us to start and stop the data collection and to label the activity being performed. The application records information at $20Hz$ (every $50ms$) from the accelerometer, gyroscope, and barometer sensors (further description shown in Table 3.1). This sampling rate has shown [27] [50] to be appropiate for this kind of systems.

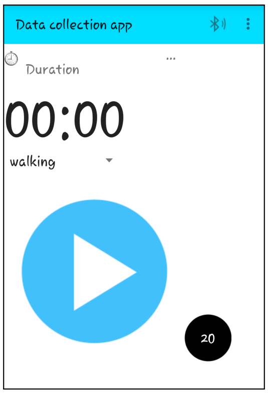


Figure 3.2: *Data collection app* GUI.

It is important to mention that the Sensor API from Android allows us to set a "suggested" minimum sampling rate for triggering its sensors events. Therefore, our selected sampling rate of $20Hz$ is only a suggested sampling rate, due to Android can alter this delay according to different criterias (e.g. low battery level). Nonetheless, as recomended in [56], we have used the timestamps associated with each sensor event to match the sampling rate over several events to $20Hz$.

Sensor	Description	Common Uses
Accelerometer	Measures the acceleration force in $\frac{m}{s^2}$ that is applied to a device on all three physical axes (x, y, and z).	Motion detection (shake, tilt, etc.)
Gyroscope	Measures a device's rate of rotation in $\frac{rad}{s}$ around each of the three physical axes (x, y, and z).	Rotation detection (spin, turn, etc.)
Barometer	Measures the ambient air pressure in hPa or $mbar$.	Monitoring air pressure changes

Table 3.1: Smartphone's sensors overview. Adapted from [56].

There is a small period of time since the user starts recording until he places the smartphone in his pocket, and also a time since the smartphone is removed from the pocket until the user stops recording. Therefore, the activity label around these times may not correspond to the actual activity performed.

To minimize mislabeling and to handle transitions between different activities without the need for stopping the recording or the help of an external observer, a second android application was developed (*Activity labeling app*) (see Fig. 3.3). This application communicates via Bluetooth any change in the activity the user is doing to the *data collection app* running on the smartphone carried in the pocket (or armband).

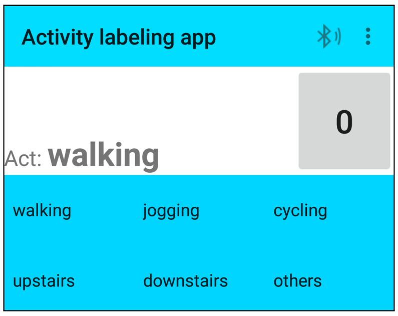


Figure 3.3: *Activity labeling app* GUI.

The *Data labeling app* was executed on a LG Optimus L3 II E430 smartphone with Android Jelly Bean version 4.1.2. The small dimensions of this device (102.6 x 61.1 x 11.9mm) make it easy to carry on the user's hands while performing common dailiy activities. We collected approximately 4 hours of data (322681 records) during 4 days, described in detail in Table 3.2.

Activity	Records	Time	Percentage
Walking	87630	1 hour 12 min	27.16%
Jogging	48187	40 min	14.93%
Cycling	37287	31 min	11.56%
Upstairs	9625	8 min	2.98%
Downstairs	9921	8 min	3.07%
Others	130031	1 hour 48 min	40.30%
Total	**322681**	**4 hours 28 min**	**100%**

Table 3.2: Data collection description.

3.2 Feature Generation

Classification algorithms cannot be directly applied to raw time-series data [27], due to they don't provide necessary details from the data. Hence, features to summarize the sensing data were generated by using a non-overlapping sliding window of five seconds, which we and other works [37] [50] [57], consider it enough time to detect patterns in the movements performed. Thus, at a sampling rate of $20Hz$, each window represents 100 samples of data.

Different time-domain features (e.g. *mean*, *max*, *min*, *standard deviation*, *variance*, *correlation*, etc.) and frequency-domain features (e.g. *Fast Fourier Transform (FFT) spectral energy*, *entropy*, *log of FFT*, etc.) have been used across most of the literature mentioned in Chapter 2. Nevertheless, with the aim to provide an energy efficient solution to improve the system's usability, we decided to extract from each window only three basic time-domain features which are computationally less expensive as mentioned in [37] [48], but critical for a good activity recognition performance, namely: *mean*, *standard deviation*, and *correlation coefficient*.

The *mean*, denoted by $\bar{x}$, is the average value over a group of observations. It has shown [58] [59] to be of importance for the accurate inference of certain activities. For example, the average acceleration for jogging is expected to be greater than for walking. Therefore, we could easily distinguish between these two activities.
Let n denote the number of samples on each five seconds window (i.e. 100 samples) and let x_1 denote the first observation on that window, x_2 the second observation, and so on up to x_n. Then $\bar{x}$, is given by:

$$\bar{x} = \frac{x_1 + x_2 + ... + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.1}$$

The *standard deviation*, denoted by s, measures how spread out the data is. Hence, it allows us to differentiate between activities based on their acceleration or rotation rate ranges. This is considered one of the most important features for the activity classification task, either by measuring the information gain [37] or by examining the effect that dropping an attribute has on the classification accuracy [57] (i.e. if the standard deviation is not considered as an activity feature then the accurcy will decrease significatively). Thus, s is calculated as:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{3.2}$$

The *correlation coefficient*, denoted by r, is a measure of the strength of the linear relationship between two numerical variables. It has demonstrated to be useful for improving identification of activities that require movement of multiple body parts [60] [61] and for distinguishing among activities that involve translations in only one dimension [30] [57] (e.g. *walking/jogging* vs. *climbing stairs*). For variables x and y:

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right) \tag{3.3}$$

Therefore, a total of eighteen features were generated on each window, divided as follows:
-*Mean* acceleration for each accelerometer axis (3 features).
-*Standard deviation* for each acceleromter axis (3 features).
-*Correlation coefficient* between all pairwise combinations of accelerometer axes. i.e. xy, xz, yz (3 features).
-*Mean* rotation rate for each gyroscope axis (3 features).
-*Standard deviation* for each gyroscope axis (3 features).
-*Correlation coefficient* between all pairwise combinations of gyroscope axes (3 features).

3.3 Activity Classification

Across the literature, authors have experimented with many different base-level classifiers, meta-level classifiers [57], and threshold-base algorithms [7] [18] [24] in order to obtain the highest possible recognition accuracy. Most of them [22] [30] [37] [51] have concluded that the Decision Tree classifier (with its default parameters) was the best classifier compared

to other important standard classifiers such as K-Nearest Neighbour (KNN), Naive Bayes, or Support Vector Machine (SVM). However, in other studies [19] [49], SVM has shown a better performance. Based on those conclusions we decided to test the performance of our system under these two machine learning classifiers.
Most of prior related works have used the popular suite of machine learning algorithm Weka [62]. However, similarly as in [33], we decided to use the Python Scikit-learn machine learning library [63] as the source of the activity recognition classifiers.

The default parameters values for the SVM classifier [64] are: *C=1.0, kernel='rbf', degree=3, gamma='auto', coef0=0.0, shrinking=True, probability=False, tol = 0.001, cache_size=200, class_weight=None, verbose=False, max_iter=-1, decision_function_shape = None, random_state = None.* The other possible values for the kernel parameter are: 'linear' and 'poly'. Initially, aiming for simpler solutions in order to reduce the overhead of our system, we tested the Linear SVM classifier. Thus, we set the parameter kernel='linear', the other parameters remained with their default values.
In a similar way, we used the Decision Tree classifier [65] with its default parameters, which are: *criterion = 'gini', splitter = 'best', max_depth = None, min_samples_split =2, min_samples_leaf =1, min_weight_fraction_leaf=0.0, max_features=None, random_state = None, max_leaf_nodes = None, min_impurity_split = 1e-07, class_weight=None, presort = False.*

In order to evaluate the accuracy of our prediction models we applied a 10-fold cross-validation over the collected data (*Setting 1*) and took the average of the accuracies on each fold. Lets denote $\hat{y}_i$ as the predicted value of the i-th sample, and y_i as the corresponding true value, then the accuracy over the total number of samples n is defined as:

$$accuracy(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} 1(\hat{y}_i = y_i) \tag{3.4}$$

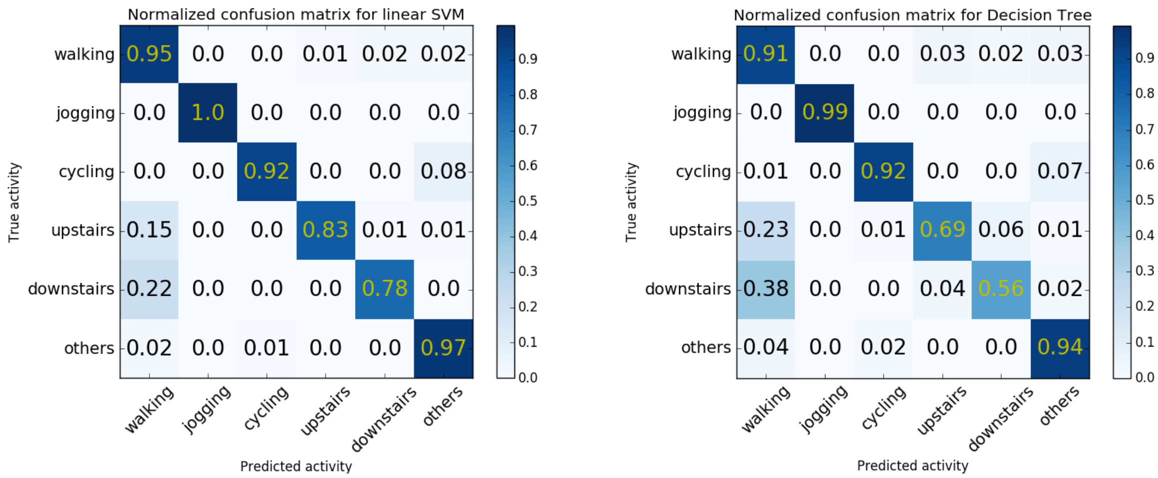
The accuracy formula described above, is the accuracy score metric provided by the Python library. However, as we can note, this metric is not able to handle class imbalance. Therefore, activities with more samples will have more weight than activities with less samples, and consequently, this value will not reflect with precision how our system behaves with activities such as upstairs and downstairs. These two activities has shown in prior studies difficulty on their identification, and therefore it is important for us to have a metric able to measure the system performance with unbalanced activities. Hence, we decided to use the average of the diagonal values from the normalized confusion matrix as our accuracy metric.

After run our classifiers according to *Setting 1*, the average accuracies were 0.93 ($\pm$0.03) and 0.87 ($\pm$0.05) for the SVM and Decision Tree classifiers respectively. The accuracy on the first five folds is detailed in Table 3.3. The class imbalance was handled by using the Stratified K-Folds cross-validation iterator from Python Scikit-learn package, where the folds are generated by preserving the percentage of samples for each class. Additionally, windows with more than one activity performed were labeled according to the major activity in that set.

Fold #	**SVM accuracy**	**Decision Trees accuracy**
1	0.9322	0.8584
2	0.9237	0.9216
3	0.9150	0.8742
4	0.9445	0.8093
5	0.8945	0.8825

Table 3.3: Model evaluation under *Setting 1*.

By analyzing results from the SVM, which is the classifier with highest accuracy, we found that the activity with the best overall accuracy was *jogging* with 1.00($\pm$0.01), then *walking* with 0.95 ($\pm$0.06), *cycling* with 0.93 ($\pm$0.08), *downstairs* with 0.90 ($\pm$0.09), and finally *upstairs* with 0.87 ($\pm$0.09). From the confusion matrices of the first five folds of *Setting 1* (see Fig. 3.5) we were able to identify that *upstairs* and *downstairs* are often mislabeled as *walking*. This mislabeling, also found in previous works [27] [32], was expected due to the similarity in the movements. Moreover, we also found that *cycling* was sometimes mislabeled as *others*. This confusion is probably caused due to we are labeling the activity as *cycling* even when the user is not necessarily pedaling, but still moving. Furthermore, we are labeling as *others* when the user stops the bike and is waiting to move again. Then, in both scenarios the leg is extended in a similar way, thus the smartphone is in the same position.



(a) Linear SVM

(b) Decision Tree

Figure 3.4: Confusion matrices from model evaluation under *Setting 2*.

Another technique to evaluate our models (*Setting 2*) consisted of using the data collected to train the models, and a different data collection carried out during three days (see Table 3.4) was used to test the models. We evaluated the models according to the same labeling criteria used in *Setting 1* (label the activity of the window with the most frequent activity in that window). The accuracy obtained with this setting was 0.9076 and 0.8338 for SVM and Decision Tree classifiers respectively. Again, SVM shows a better performance than the Decision Tree classifer. As we can note in Fig. 3.4, the main difference between these two classifiers occurs in the identification of *upstairs* and *downstairs* activities, where the SVM outperforms the Decision Tree classifier. Furthermore, similarly as with *Setting 1* evaluation, there are mislabelings on *upstairs* and *downstairs* with *walking*, and in less proportion between *cycling* and *others* . We can also note that with *Setting 2*, *downstairs* have more misclassifications than in *Setting 1*, even more than *upstairs*. From now on we only use *Setting 2* due to we consider it reflects better a real scenario.

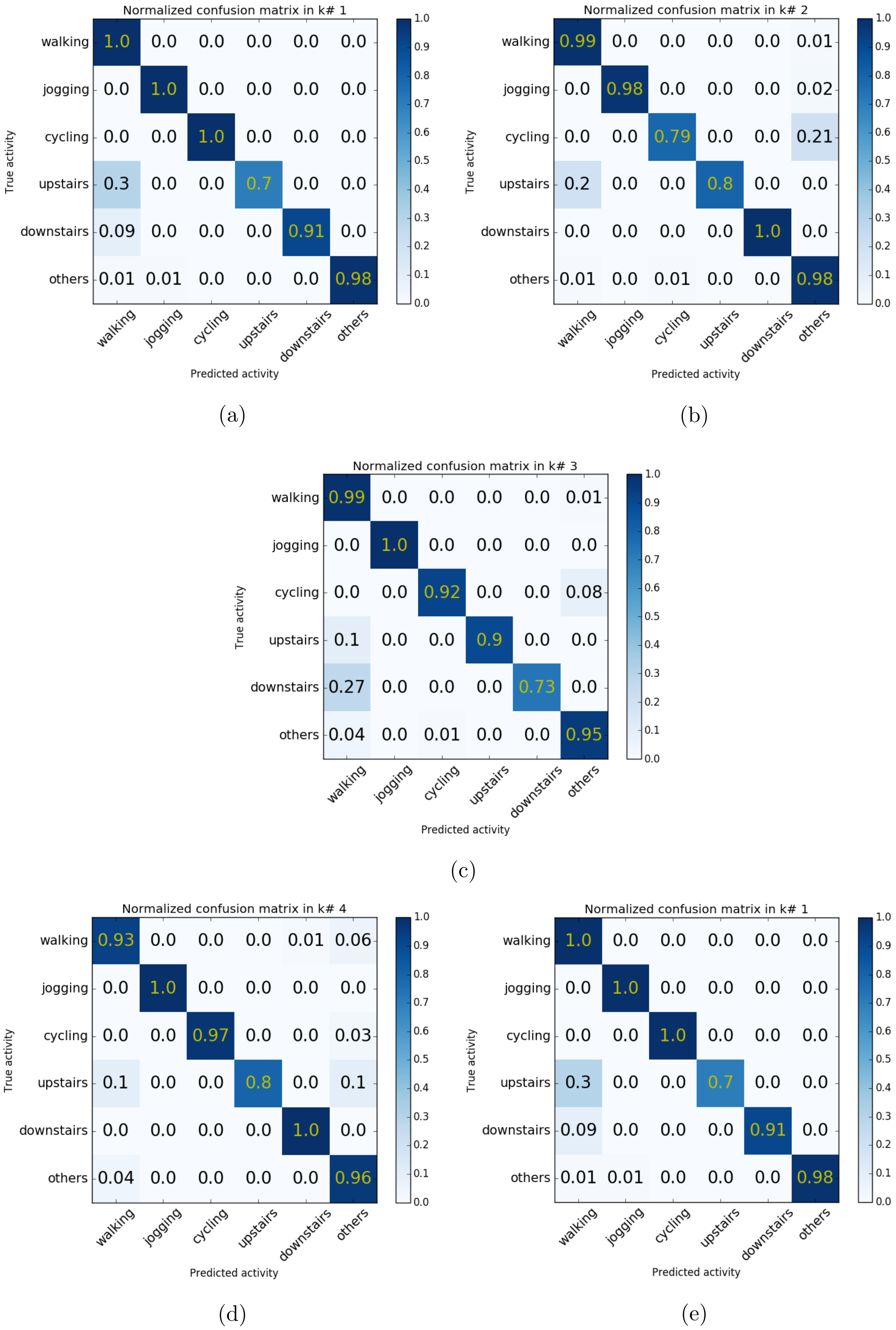


Figure 3.5: Confusion matrices from model evaluation under *Setting 1*.

Activity	Records	Time	Percentage
Walking	84739	1 hour 10 min	29.81%
Jogging	53368	44 min	18.77%
Cycling	40100	33 min	14.10%
Upstairs	10457	9 min	3.67%
Downstairs	9002	7 min	3.16%
Others	86618	1 hour 12 min	30.47%
Total	**284284**	**4 hours**	**100%**

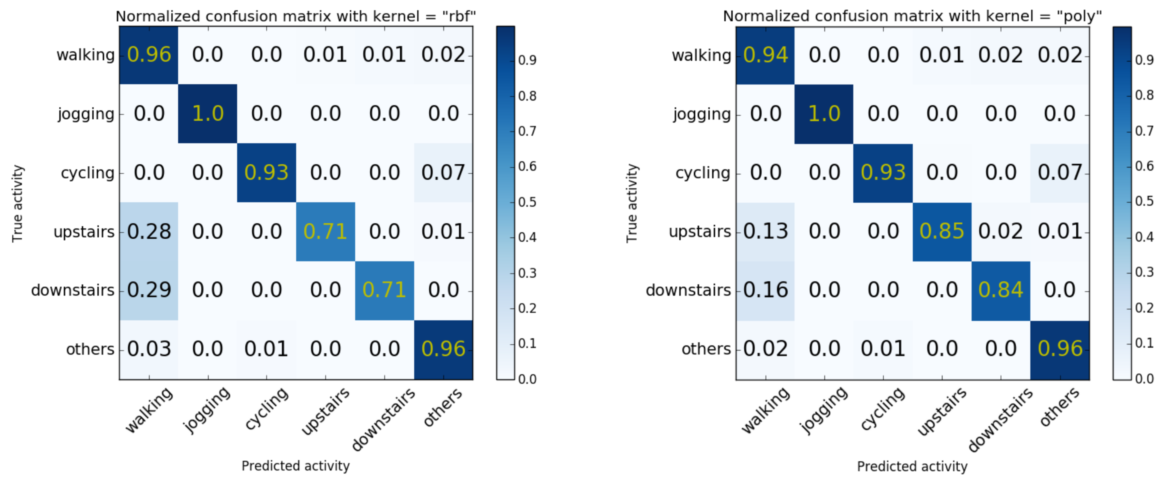
Table 3.4: Description of data collection used as test set on *Setting 2*.

We experimented with non-linear approaches of the SVM classifier to observe how our system behaves, and compared it with the linear approach. In order to do so, we set two possible values to our kernel parameter: 'rbf' or 'poly'.

Kernel	Accuracy
linear	0.9076
rbf	0.8778
poly	0.9191

Table 3.5: Comparison of SVM classifier performance with different kernel configurations.

Results from Table 3.5 shows that the accuracy of the model when kernel is set to 'poly' is the best one overall. In contrast, when kernel = 'rbf' the confusion of *upstairs* and *downstairs* with *walking* increases (see in Fig. 3.6). Even though the difference between the Linear SVM classifier and the nonlinear SVM classifier with kernel='poly' is minimal, we tried to increase this difference by testing our model with different C and gamma parameters from our classifier. C, trades off mislabeling against simplicity of the decision surface, therefore, low C values provide a smooth decision surface, and high C values classify all training examples correctly. Gamma, defines how much influence a single training example has, therefore, larger gamma values means the closer other examples must be to be affected.



(a) kernel='rbf' (b) kernel='poly'

Figure 3.6: Confusion matrices from model evaluation with nonlinear SVM classifiers

From the parameter tuning (see Fig. 3.7) the best C and gamma values were C=0.1 and gamma=0.1. The classification accuracy with these parameters under *Setting 2* was 0.9213 (less than 1% increment). The accuracies for each activity were: *walking* 0.95, *jogging* 1.0, *cycling* 0.93, *upstairs* 0.84, and *downstairs* 0.85. As we can note *walking*, *jogging* and *cycling* are easily identifiable. Nonetheless, *upstairs* and *downstairs* are still the hardest activities to recognize due to their similarity with *walking*.

The activity recognition accuracy with the Linear SVM was approximately 0.91. Therefore, the increment of 2% obtained with the best model of the nonlinear SVM approach is not significative. Moreover, due to the system will run on a smartphone, we consider appropiate to use a simpler technique. Hence, we selected the Linear SVM as our activity classifier.

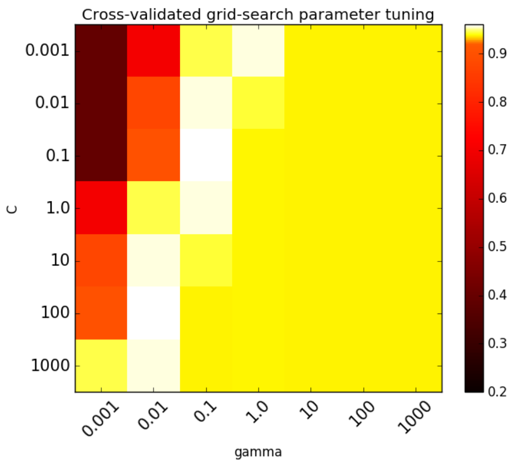


Figure 3.7: Cross-validated grid-search parameter tunning for nonlinear SVM classifier with kernel='poly'.

3.4 Gyroscope sensor importance

Prior works in activity recognition used the built-in accelerometer sensor as the main source of information to identifiy movements. In this study we included the gyroscope sensor to provide an aid to the accelerometer. To analyze how useful the gyroscope is, we tested our system by using only features from the accelerometer and compared its results with the results obtained previously with the combination of both sensors.

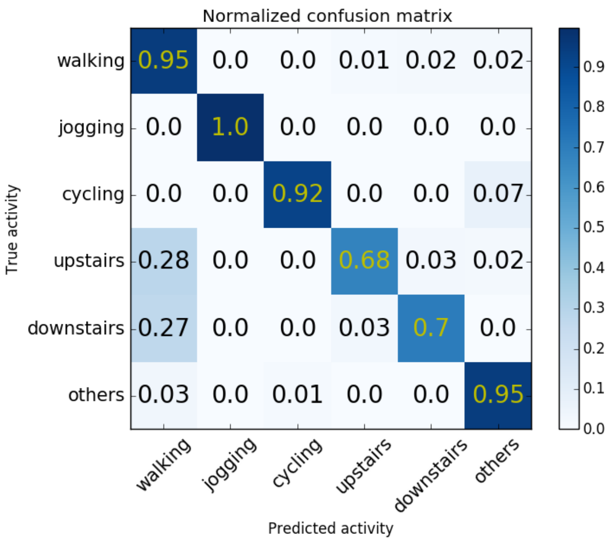


Figure 3.8: Confusion matrix for activity recogniton by using only the accelerometer sensor.

Therefore, we only generated nine features from each window, and used them to create our classification model. The accuracy obtained with the Linear SVM classifier by only using the accelerometer features was 0.8650, which is still good, but with both sensors we obtained 0.9076. Thus, the addition of the gyroscope sensor provided an increment of approximately 4%. Also, if we analyze the confusion matrix (see Fig. 3.8), we can note that the gyroscope was mostly helpful to reduce the mislabeling of *upstairs* and *downstairs* with *walking*, due to by only using the accelerometer the average accuracy of the *stairs* activities is 0.69 and with both sensors was in average 0.81. Hence, with the gyroscope we increased by 12% the accuracy in the identification of these activities.

Chapter 4

Caloric expenditure estimation

The calories burned during an activity are strongly related to the amount of physical activity performed. Therefore, this is considered a useful metric to measure the fitness level of an individual. There exists several accurate approaches to estimate the calories burned. Direct calorimetry [66] measures the heat output of the body to quantify the energy expenditure. Another reliable method is Doubly-labeled water [67], where the amount of exhaled carbon dioxide CO_2 provides a close approximation of the calories expended during an activity. Lastly, the Indirect calorimetry approach [68] estimates the calories burned by measuring the volume of oxygen consumed VO_2 while performing an activity. However, despite being effective, these methods are not suitable for everyday life as they are constrained to laboratory settings.

There also exists simpler but less accurate approaches such as MET, PAL and PAR units that relate type of activity to an index in order to quantify the energy expenditure. However, they do not consider the user's effort during an activity. Finally, there are metabolic equations published by the ACSM, which through certain parameteres provide estimations of the VO_2. This method is more accurate than using indexes due to it considers parameters such as weight, number of steps, speed and grade levels, which help to provide a more real approximation regarding the user's effort and consequently the calories burned.

In this chapter we explain how we use the ACSM's metabolic equations to estimate the VO_2 and the calories burned on the set of five identified activities namely *walking*, *jogging*, *cycling*, *upstairs*, and *downstairs*. Furthermore, we describe the different data collection, experiments and assumptions that we made to exploit the smartphone's sensing capabilities in order to provide values to the *number of steps*, *speed*, *grade* and *work rate* parameters neccesary for the metabolic equations.

4.1 ACSM's metabolic equations

All caloric expenditure in the body can usually be regarded as being aerobic and thus quantifiable in terms of oxygen consumption VO_2 [69], which is a numerical measurement that indicates the volume of oxygen consumed while performing an activity. This relation is largely due to more than 95% of the energy expended by the body is derived from oxidation of nutrients [70].

The gold standard for measuring VO_2 is through a test where you are hooked up to a breathing mask while undergoing a progressively more difficult exercise on a treadmill. However, in order to estimate caloric expenditure from the motion activities such as *walking*, *jogging*, *upstairs*, *downstairs*, and *cycling*, we have calculated the VO_2 by using the equations (see Table 4.1) provided by the American College of Sports Medicine (ACSM) [71].

Where,

$$VO_2 = \text{Resting Component} + \text{Horizontal Component} + \text{Vertical Component}$$

Finally, we can estimate the calories burned by multiplying VO_2 in $\frac{L}{min}$ by a factor of $5.01\frac{kcal}{L}$ (for every liter of oxygen consumed approximately 5 $Kcal$ are burned [71]).

$$\text{Calories} = VO_2\ \frac{L}{min} * 5.01\frac{kcal}{L} = \frac{kcal}{min}$$

The equations for VO_2 estimation require information about the type of activity being performed, user weight, number of steps, speed of movement, grade at which the user is moving, step height and work rate. Only the weight needs to be provided by the user, the other information can be calculated from the smartphone's sensors. Furthermore, it is important to mention that since walking upstairs is a more strenuous activity than walking downstairs, we are not considering the resistance component (vertical component) equations in the VO_2 estimation of this stepping activity (i.e. *downstairs*).

Activity	**Resting Component**	**Horizontal Component**	**Vertical Component / Resistance**	**Limitations**
Walking	3.5	0.1 x *speed*[a]	1.8 x *speed*[a] x *grade*[b]	Most accurate for speeds of 3.0–5.9 $km.h^{-1}$ (50-100 $m.min^{-1}$)
Running	3.5	0.2 x *speed*[a]	0.9 x *speed*[a] x *grade*[b]	Most accurate for speeds >8 $km.h^{-1}$ (134 $m.min^{-1}$)
Stepping	3.5	0.2 x $steps.min^{-1}$	1.33 x (1.8 x *step height*[c] x $steps.min^{-1}$)	Most accurate for stepping rates of 12-30 $steps.min^{-1}$
Leg cycling	3.5	3.5	(1.8 x *work rate*[d]) / *body mass*[e]	Most accurate for work rates of 300-1.200 $kg.m.min^{-1}$ (50-200 W)

[a] *speed* in $m.min^{-1}$.

[b] *grade* is percent grade expressed in decimal format (e.g., 10% = 0.10).

[c] *step* height in m.

[d] *work rate* in kilogram meters per minute ($kg.m.min^{-1}$) is calculated as resistance (kg) x distance per revolution of flywheel x pedal frequency per minute. Note: Distance per revolution is $6m$ for Monark leg ergometer, $3m$ for the Tunturi and BodyGuard ergometers.

[e] *body mass* in kg.

Table 4.1: Metabolic Equations for Gross VO_2 [$mL.kg^{-1}.min^{-1}$] during common physical activities. Adapted from [71]

4.2 Data collection

We need to estimate the number of steps, speed, grade, step height and work rate of a user in order to be able to use the ACSM's equations. Therefore, we performed different experiments in order to prove that the smartphone's embedded sensors are capable of providing accurate estimations of these parameters and consequently accurate caloric expenditure estimations as well. All the collected information contains readings from the accelerometer, gyroscope and barometer sensors. Furthermore, for each experiment different data collections were conducted as described below.

4.2.1 Steps data collection

To compute the number of steps, we gathered different records from each activity (i.e. *walking*, *jogging*, *cycling*, *upstairs*, and *downstairs*). These records were used as our ground truth for this experiment due to they contain a field where the real step count was stored. To record the real step count, we added a step counter button in the *Activity labeling app* as shown in Fig. 3.3. Therefore, any time the user performs a step, the button is pressed and the step is recorded. Besides, in the case of *cycling*, every pedaling was considered a "step". We used 70% of these records (*steps training dataset*) for parameter tuning and 30% to test our algorithm (*steps test dataset*).

4.2.2 Step height data collection

In order to estimate the step height in the stairs activities, we gathered data while the user climbed stairs at three different buildings (*step height dataset*), going from the lowest level to the highest level of the respective building. The total number of steps was provided by using the step counter button from the *Activity labeling app*. Besides, we also manually annotated the total number of stairs and the stair step height from each building.

4.2.3 Grade data collection

For the grade computation, we collected sensing data while the user perfoms five walks at five different routes. Records from three routes were used for parameter tuning purposes (*grade training dataset*) and the data from the other two routes was used to test our algorithm (*grade test dataset*).

4.2.4 Work rate data collection

As we will see later in Section 4.6, the *cycling* speed is an important parameter to estimate the *work rate.* Thus, we collected data from the smartphone's sensors while the user rode a bike at four different routes (*cycling dataset*). We used as ground truth for our tests a bicycle computer (see Fig. 4.1) that will gives us the *distance traveled*, *average speed*, and *maximum speed* while cycling.

Figure 4.1: Suaoki Bicycle Computer. Image from [72]

4.3 Steps detection

To estimate the VO_2 for the stepping activities (*upstairs* and *downstairs*), we need to count the number of steps given per minute (i.e. $steps.min^{-1}$). However, as we will see later, counting the number of steps is also important for the VO_2 estimation of the remaining activities (*walking*, *jogging*, and *cycling*). Therefore, we have developed an algorithm that uses a zero crossing technique to detect steps in all the activities.

Previous works [55] [73] [74] have developed similar algorithms by reading accelerometer data. Nonetheless, we decided to use the gyroscope sensor data for this task. The gyroscope sensor measures the rate of rotation in $\frac{rad}{s}$ around a device's x, y, and *z-axis* (see Fig. 4.2), and as mentioned before, it has proved a better performance than the accelerometer when the smartphone is carried on pant's pocket. In addition, the rotation rate data comes already with drift compensation. Therefore, the signals from this sensor have less noise than the signals from the accelerometer sensor.

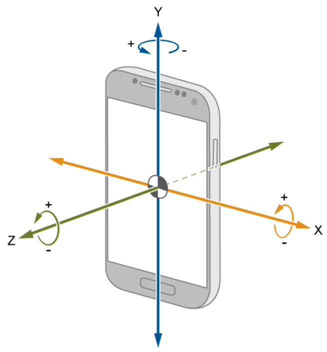


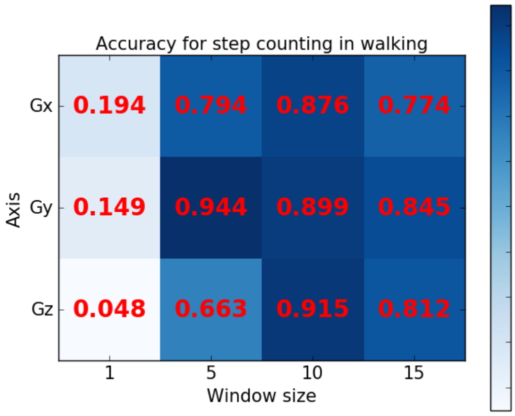
Figure 4.2: Gyroscope sensor axes. Image from [75]

Despite small noise still existing in the gyroscope signals, we implemented an average filter to act like a low-pass filter in order to achieve more accurate estimations. The average filter uses an overlapping sliding window technique where the size of the window and data source (gyroscope axis signal) depends on the activity.

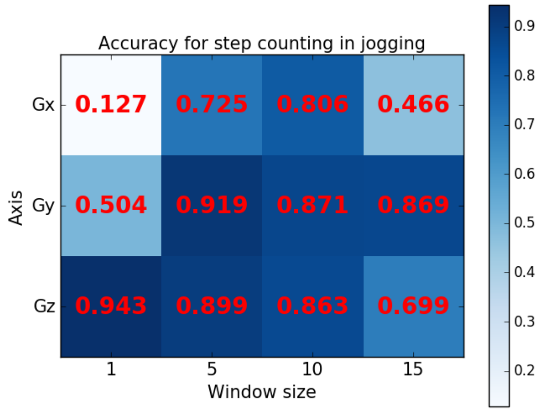
A grid-search parameter-tuning over the *steps traning dataset* was performed (see Fig. 4.3) in order to identify the best window size and gyroscope axis combination for each activity. We have denoted to the gyroscope *x*, *y*, and *z-axis* as *Gx*, *Gy*, and *Gz* respectively. Results are detailed in Table 4.2.

Activity	**Axis**	**Window size (# samples)**	**Accuracy**
Walking	*Gy*	5	0.944 (±0.04)
Jogging	*Gz*	1	0.943 (±0.08)
Cycling	*Gy*	10	0.950 (±0.05)
Upstairs	*Gx*	5	0.896 (±0.09)
Downstairs	*Gx*	10	0.896 (±0.09)

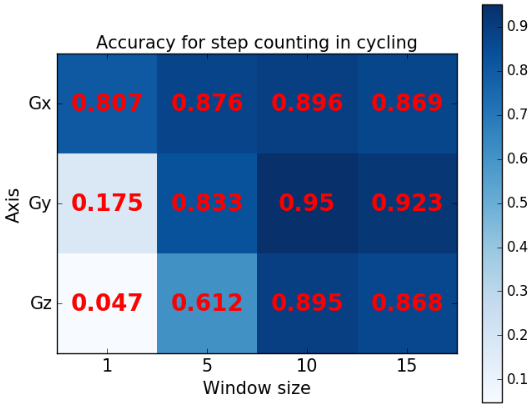
Table 4.2: Gyroscope axis and window size values for best step estimation based on activity.



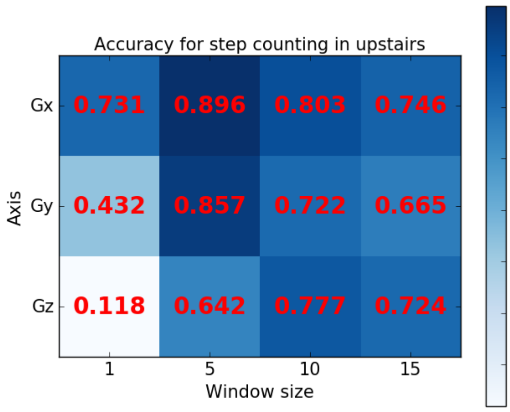
(a)



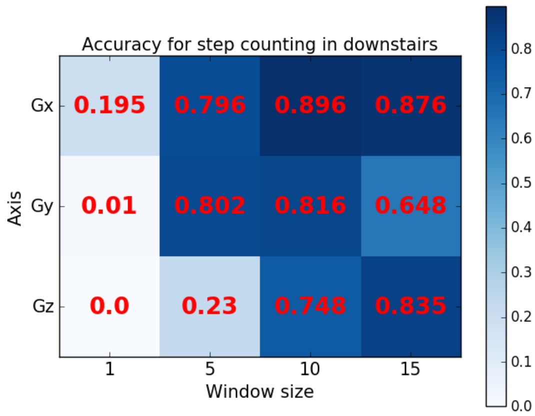
(b)



(c)



(d)



(e)

Figure 4.3: Parameter tuning for step detection on each activity.

Most of the activities had a step detection accuracy above 90%, except for *upstairs* and *downstairs* which accuracies are the lowest and with highest standard deviation. Hence, we tested our best parameters in *steps test dataset*. Results are shown in Table 4.3.

Activity	**Accuracy**
Walking	0.836 ($\pm$0.09)
Jogging	0.889 ($\pm$0.05)
Cycling	0.866 ($\pm$0.10)
Upstairs	0.844 ($\pm$0.18)
Downstairs	0.824 ($\pm$0.08)

Table 4.3: Step detection accuracies in *steps test dataset*.

Despite having good results overall in our tests, we can still see some variance in the accuracies of the *walking*, *cycling*, and *downstairs* activity; and an even higher variance in the accuracy of the *upstairs* activity.

In the case of the *walking* activity we found that despite smoothing the gyroscope signal with the average filter, the remaining noise was causing some false zero crossings (false steps) that reduce the accuracy of the step detection algorithm.

In Fig. 4.4 we plot six seconds of walking data, highlighting with blue color the zero crossings on each axis. It is worth to mention that this data is already smoothed by using a window of 5 samples, and besides the real number of steps for this record was 135. The number of steps detected on each gyroscope axis were 206, 144, and 214 for Gx, Gy, and Gz respectively. As we can note, Gy provided us the best estimation, which was expected according to the parameter tuning results showed above. However, we can observe that the signal in Gx have some false zero crossings due to small peaks generated before the higher peaks (which are the steps). Therefore, we included into our step detection algorithm a minimum time threshold ($minTime$) between zero crossings to minimize the effect of these false readings.

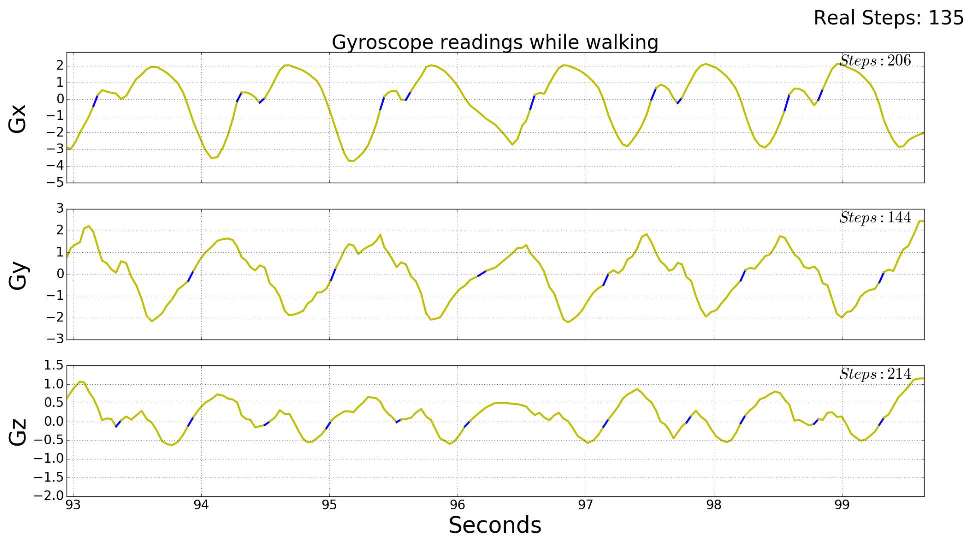


Figure 4.4: False zero crossings (blue lines) in gyroscope signals.

We set a minimum time threshold between zero crossings of 0.5 seconds. We consider this time engouh to avoid the double detection in consecutives zero crossings. As we can note in Fig. 4.5, the number of false zero crossings was reduced, and its effect can be appreciated with a more approximated estimation of the number of steps in *Gx*.

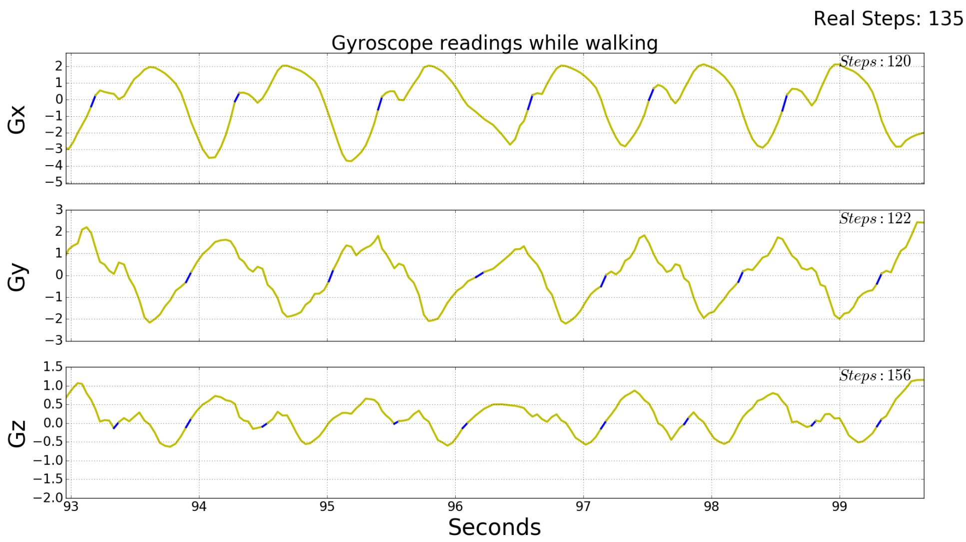


Figure 4.5: Falso zero crossings reduced by setting $minTime$=0.5 seconds.

With the additional threshold setting we perfomed a new grid-search parameter-tuning in *steps training dataset* looking for the best parameters' combination per activity. As we can note in Fig. 4.6, the accuracy results for *walking* increased, especially in *Gx*. Gyroscope axis *Gy* with an average filter of 5 samples continue being the best parameter combination. Nevertheless, *Gx* has also high accuracy values and looks like the average filter for smoothing the signal is less important due to even with a window size of 1 (no smoothing), *Gx* and *Gy* achieved high accuracy values 0.905 and 0.851 respectively.

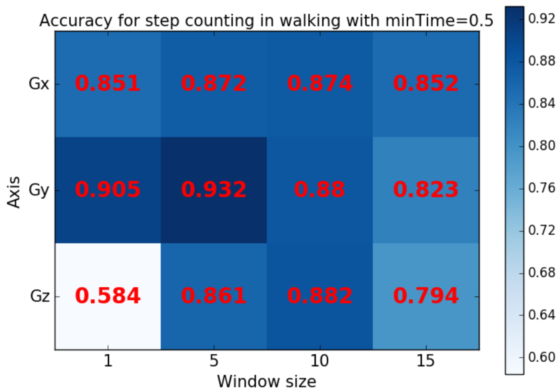


Figure 4.6: Parameter tuning for *walking* by using *minTime*=0.5 seconds.

After testing the best parameters for *walking* (*Gy* axis and average filter of 5 samples) with the additional time threshold of 0.5 seconds in *steps test dataset*, we obtained an accuracy of 0.937 (±0.03) with is much better compared with the previously *walking* step detection accuracy of 0.836 (±0.09).

The other activities where the smartphone is carried in the same position as for *walking* (in the pant's front left pocket) are *cycling*, *upstairs*, and *downstairs*. Hence, as we did with *walking*, we performed another grid-search paramter tuning (using *minTime*=0.5 seconds) over these activites.

Similarly as in the *walking* activity, the accuracy values for the *cycling* step detection in *Gx* have improved. Results from the grid-search parameter tuning shown in Fig. 4.7, indicate us that *Gx* with window size of 1 (no smoothing), achieved the highest accuracy with 0.961(±0.04). However, as it is important to smooth the gyroscope signals in case of noisy readings. We selected the *Gy* axis with window size of 5 as our best parameters, which in the parameter tuning achieved an accuracy of 0.945(±0.06).

After testing the best parameters for cycling (*Gy* with average filter of 5 samples) with

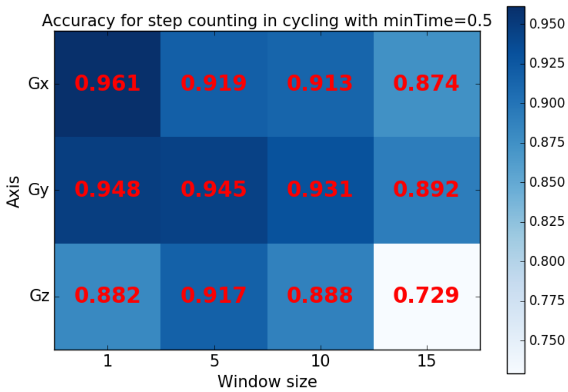
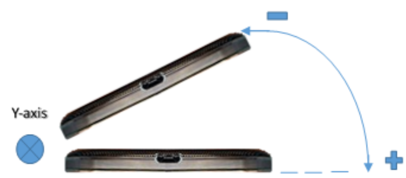


Figure 4.7: Parameter tuning for *cycling* by using *minTime*=0.5 seconds.

the additional time threshold of 0.5 seconds in our *steps test dataset*, we obtained an accuracy of 0.914(±0.06), which is better compared with the previous *cycling* step detection accuracy of 0.866(±0.10).

With the addition of the time threshold between zero crossings, we have effectively improved the accuracy and reduced the variance of the walking and cycling activities. We would expect from *Gx* to be the best axis at detecting steps, due to as shown in Fig. 4.8a, the rotation in this axis follows the leg forward movement. Nevertheless, from the parameter tuning results, *Gy* seems to be the most representative axis for step detection despite each step only generates a small rotation (see Fig. 4.8b) due to the leg muscles' contraction.

(a) Rotation in *x-axis*

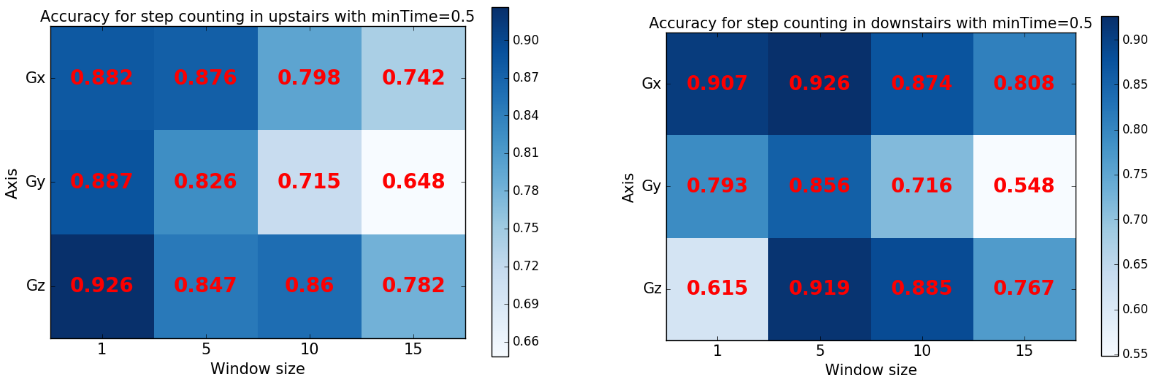


(b) Rotation in *y-axis*

Figure 4.8: Smartphone rotation while walking.

However, by keeping the best window size from both activities (5 samples), but with the *Gx* instead of *Gy* as best axis, results against *steps test dataset* are also high, 0.88(±0.04) for walking and 0.904(±0.06) for cycling. Therefore, we could conclude that both axes will perform quite well in the step detection task and that smoothing the data before analysis is an important process to get accurate results.

In the *downstairs* activity, the implementation of the time threshold of 0.5 seconds, as expected, increased the accuracy of the step detection in Gx, it also reduced the window size (from 10 samples to 5 samples) of the average filter as well. The new best parameters for *downstairs* after parameter tunnig with *steps training dataset* were Gx with window size of 5 samples (see Fig. 4.9b). Nevertheless, a sligthly different situation occured with the *upstairs* activity. The accuracy of the step detection increased overall and the window size was reduced to point of no need of the filter as shown in Fig. 4.9a. Nonetheless, the gyroscope axis considered the most significant for this activity changed from Gx to Gz. Furthermore, as we can note in Fig. 4.9b, the second best parameters' combination for detecting downstairs steps was Gz with window size of 5 samples. Therefore, we could conclude that Gz is also an important gyroscope axis for *walking stairs* activities.



(a) Upstairs activity

(b) Downstairs activity

Figure 4.9: Parameter tuning for walking stairs activities by using *minTime*=0.5 seconds.

We tested these new best parameters for detecting steps at *downstairs* (Gx with window size of 5 samples) and *upstairs* (Gz with window size of 1) on *steps test dataset*. We obtained an accuracy of 0.897(±0.06) at detecting *downstairs* steps and 0.825(±0.05) for *upstairs* steps. Therefore, comparing these results with the accuracy results obtained without the time threshold implementation, we have achieved an improvement.

Despite having achieved a good step detection accuracy on upstairs by using the raw Gz data (window size of 1), we decided to test it with another parameters' combination. We selected the Gx axis with window size of 5 due to it was the prior best parameters' combination for upstairs step detection (before including minTime threshold), it showed a good performance on parameter tuning (0.876), and lastly because we consider important the signal smoothing process in case of noisy data. We obtained an slightly better accuracy of 0.842, but with higher standard deviation (±0.19). Hence, *Gz* has shown to be more consistent than *Gx* at detecting *upstairs'* steps.

A summary of the accuracy results and best parameters' combination used with the implementation of the minimum time threshold of 0.5 seconds for step detection on all the activities where the smartphone is carried in the front left pocket is shown in Table 4.4.

Activity	**Axis**	**Window size (# samples)**	**Accuracy**
Walking	*Gy*	5	0.937 (±0.03)
Cycling	*Gy*	5	0.914 (±0.06)
Upstairs	*Gz*	1	0.825 (±0.05)
Downstairs	*Gx*	5	0.897 (±0.06)

Table 4.4: Best parameters and step detection accuracies in *steps test dataset* by using minTime=0.5 seconds.

For *jogging*, the smartphone is carried in an armband, and the device rotation in the *z-axis* (*Gz*) follows the movement described in Fig. 4.10. Therefore, the expected rotation axis to be the most representative for step detection in this activiy is *Gz*.

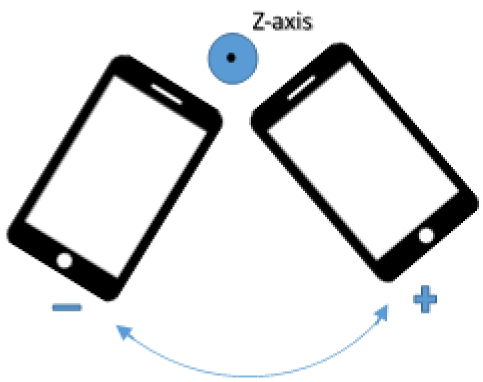


Figure 4.10: Smartphone rotation while jogging.

From parameter tuning on this activity we can note (see Fig. 4.3b) that as we expected, the *z-axis* achieved the best step detecting accuracy and even without any smoothing required (i.e. window size=1). The average filter isn't necessary for this activity due to despite the others activities where the smartphone is in the pant's pocket (which doesn't keep the smartphone always exactly in the same position), when *jogging*, the armband reduces the possible small movements and the signal has less noise. Furthermore, the effect of the minimum time threshold was also tested in the *jogging* activity, however, the accuracy result didn't improve.

We have demonstrated that it is possible with the only use of the smartphone's embedded gyroscope to estimate with an average accuracy of 86% the number of steps while performing stepping activities such as *upstairs* and *downstairs*. Hence, we are able to provide an accurate approximated value to the $steps.min^{-1}$ parameter necessary for VO_2 estimation on stepping activities. Similarly, as explained in the following sections, the step detection on the other activities (*walking*, *jogging*, and *cycling*) is also useful at estimating VO_2 for those activities.

4.4 Speed computation

Speed is an important parameter at estimating VO_2 on *walking* and *jogging* activities. In order to calculate the speed of movement on those activities, we need to know the distance the user has traveled (due to time is already known), which could be estimated easily by the GPS coordinates. However, in favor of an energy efficient solution we have decided to estimate the distance with the following equation:

$$distance = stride\ length * step\ count \tag{4.1}$$

Where the *step count* can be estimated with an 93.7% accuracy on walking and 88.9% on jogging with our step detection algorithm, and the *stride length* is calculated based on the equation from [76]:

$$stride\ length = height * k \tag{4.2}$$

Being *height* the user's height and k equals to 0.415 for men and 0.413 for women in the case of walking. And for jogging, the value of k is equal to 0.65 for men and 0.55 for women.

4.5 Grade and step height computation

The smartphone's barometer sensor has proved a great performance at identifying activities that involve vertical displacements [41] [42] . Therefore, the use of this sensor allow us to provide an accurate estimation of the grade and step height. However, despite the barometer only senses current barometric pressure in hPa, we have translated those pressure readings into a metric more related with elevation by using the following equation from [77]:

$$altitude = 44330 * (1 - (\frac{P}{Po})^{\frac{1}{5.255}}) \tag{4.3}$$

Where Po is the pressure at sea level = $1013.25hPa$, P is the pressure reading value, and *altitude* represents the value in meters of pressure P.

4.5.1 Step height estimation

Even though we could use a fixed value for the step height parameter (assuming a standarized stair step height), we decided to make some experiments in order to test the efficiency of the barometer sensor at different granularity levels. Thus, we used the data from *step height dataset*, where with the total number of stairs and stair step height information from each building, we first calculated the real elevation gain of each building (multiplying both parameters). The estimated elevation gain was calculated as the difference between the elevation of the last recorded point with the elevation from the first recorded point. However, as we do not have elevation values, we translated the barometric pressure from those two readings into altitude values by using the equation 4.3.

Results described in Table 4.5 show that the barometer is capable of estimating the elevation gain with an accuracy of 0.91(±0.04). Nonetheless, we also tested a finer level of granularity.

Building #	**Real Elevation Gain *(m)***	**Estimated Elevation Gain *(m)***	**Accuracy**
1	29.76	30.99	0.959
2	4.76	4.24	0.891
3	6.6	5.74	0.87

Table 4.5: Elevation gain estimations.

By using our step detection algorithm for upstairs activities, we estimated the step counts on each building. Moreover, with the information already obtained about the estimated elevation gain, we could estimate the stair step height and compare it with the real step height annotated from each building.

Results from Table 4.6 show us that the barometer estimated step heights with an accuracy of 0.73(±0.14). Despite not having high estimations, we need to consider that in the *upstairs* data from *step height dataset* there are also included some small *walking* periods (when going from one stair to another). Therefore, the step counts either real or estimated do not only belong to *upstairs* but in some small proportion to *walking*.

Building #	Real Step Height (cm)	Estimated Elevation Gain (m)	Estimated Step Count	Estimated Step Height (cm)	Accuracy
1	15	30.99	194	15.97	0.935
2	17	4.24	38	11.16	0.657
3	16.5	5.74	58	9.90	0.600

Table 4.6: Step height estimations.

However, due to we annotated the total number of stairs and the stair step height, we could provide a more accurate approximation about the real steps count during *climbing stairs* (*real step count upstairs-specific*). Moreover, by using the steps count provided from our *Activity label app* and compare it with *real step count upstairs-specific*, we calculated an average overestimation percentage of 18%. Hence, with this percentage we could obtain a more accurate estimation about the number of steps performed specifically while *climbing stairs* (not including steps when *walking* between stairs). Results from Table 4.7 show us that the barometer is capable of estimating with an accuracy of $0.74(\pm 0.04)$ the step height during *upstairs*. Furthermore, as we can note, by using a more approximate estimation about the step counts, we have slightly increased the accuracy and reduced by a 10% the standard deviation, providing a more consistant result.

Building #	Real Step Height (cm)	Estimated Elevation Gain (m)	Estimated Step Count	Estimated Step Count (upstairs specific)	Estimated Step Height (cm) (upstairs specific)	Accuracy
1	15	30.99	194	159	19.49	0.701
2	17	4.24	38	31	13.67	0.804
3	16.5	5.74	58	47	12.21	0.740

Table 4.7: Step height estimations not considering small *walking* periods while climbing stairs.

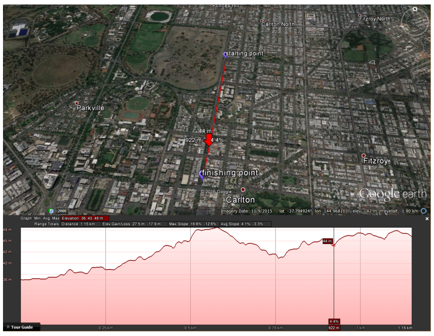
4.5.2 Grade computation

We calculated the grade necessary for the VO_2 estimation on walking and jogging activities with the following equation from [77]:

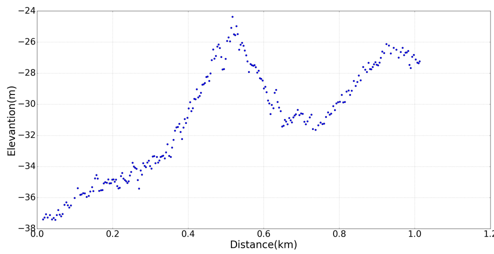
$$G_k = 100 \text{ x } (AD_k/D_k)$$

Where, $G_k[\%]$, $D_k[m]$ and $AD_k[m]$ represent the gradient, distance, and difference of elevation in the *k-th* time window, respectively. Here, $AD_k[m]$ is calculated as the difference between the first and last recorded point in the *k-th* time window. However, as those points represent pressure, we used the equation 4.3 to calculate AD_k in meters. Besides, D_k was estimated by using the equation 4.1.

By using the data from *grade training dataset* we drawed the paths of each route on Google Earth (see Fig. 4.11a), and used the elevation gain and loss obtained from the elevation profile (of each path) to set our ground truth.



(a) Google Earth's elevation profile



(b) Estimated elevation profile

Figure 4.11: Elevation profile comparison.

To estimate the elevation gain and loss we used an average filter over a non-overlapping sliding window to reduce noise, and over the resulting smoothed data we calculate elevation gain by summing up all the gains, where a gain is defined as the difference between a low point and a high point. Similarly, the elevation loss was estimated in the same way, but summing up all the losses instead of the gains. This estimation was performed for each path using different windows sizes (5, 10 and 15 seconds) and then the average accuracy with its standard deviation was calculated (see Table 4.8).

Window size (s)	**Elevation Gain Accuracy**	**Elevation Loss Accuracy**
5	0.70 ($\pm$0.07)	0.66 ($\pm$0.10)
10	0.46 ($\pm$0.22)	0.40 ($\pm$0.17)
15	0.36 ($\pm$0.20)	0.27 ($\pm$0.13)

Table 4.8: Elevation gain/loss estimations by using Google Earth.

Results from this experiment may look bad, but we need to consider that the ground truth from Google Earth elevation profile it is not a strong ground truth. Google Earth's elevation data is not high resolution, elevation accuracy in Google Earth is in the range of $\pm 30m$, which obviously is not useful for anything in which you would need accurate numbers.

Therefore, in order to use a more accurate ground truth, we used ELVIS [78], which is an elevation system (see Fig. 4.12) developed by the Australian Government. Elevation data from ELVIS has a $\pm 5m$ error, and consequently constitutes a better ground truth for our analysis.

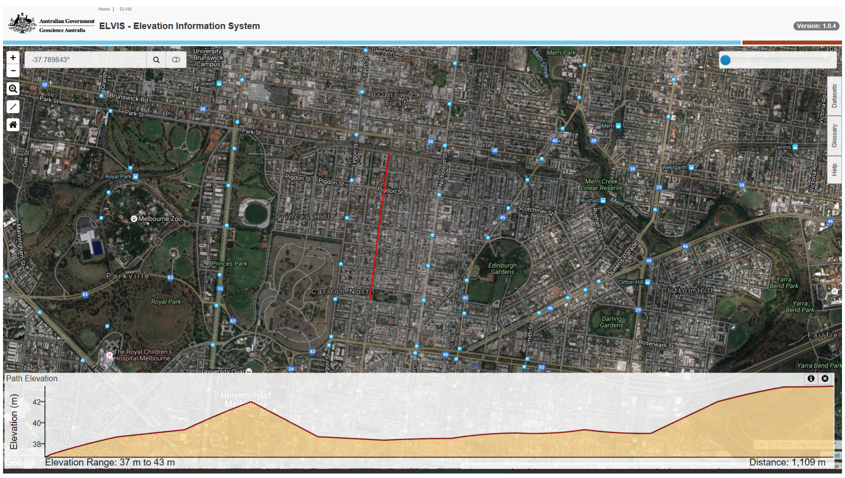


Figure 4.12: ELVIS (Elevation Information System).

We ran the same experiment described above but with the ground truth data of elevation gain and loss from ELVIS. Results from this experiment (see Table 4.9) indicated us that with a window size of 10 seconds we can get accurate results.

Window size (s)	**Elevation Gain Accuracy**	**Elevation Loss Accuracy**
5	0.33 ($\pm$0.23)	0.17 ($\pm$0.25)
10	0.86 ($\pm$0.12)	0.80 ($\pm$0.08)
15	0.64 ($\pm$0.13)	0.58 ($\pm$0.15)

Table 4.9: Elevation gain/loss estimations by using ELVIS.

We tested this window size in *grade test dataset*. For elevation gain we obtained an accuracy of 0.98 ($\pm$0.01) and for elevation loss the accuracy was 0.78 ($\pm$0.18). Despite having a very good accuracy in elevation gain, we can see some variability in the accuracy of the elevation loss. However, we should consider that the ground truth source has still some error percentage and that we only used two records as test set. Moreover, the elevation gain is the metric necessary for the grade estimation, and therefore 98% of accuracy supports highly accurate grade estimations as well.

4.6 Work rate computation

The ACSM's equation for VO_2 estimation while *cycling* requires two parameters: *work rate* and *body mass*. The *body mass* or user's weight, as mentioned before, is provided by the user. *Work rate* can be defined as:

$$work\ rate = resistance * distance\ per\ revolution * pedal\ frequency\ per\ minute \tag{4.4}$$

The *pedal frequency per minute* can be estimated with an 91% of accuracy from our step detection algorithm at *cycling*. Furthermore, the *distance per revolution* is given by the bicycle wheel's circunference, which could be estimated by knowing the diameter of the wheel (entered by the user) and multiplying it by π, or by using a standard wheel circunference value. Nevertheless, the *resistance* depends on some other forces. According to [40], *resistance*, denoted by F, can be estimated as:

$$F = F_r + F_g + F_a \tag{4.5}$$

Where,

$F_r = mgC_r$, is the rolling resistance on the bike,

$F_g = mgs$, is the component of gravity along the direction of movement, and

$F_a = \frac{1}{2}\rho(T)C_aV_a^2A$, is the force of aerodynamic drag.

From above equations; m is the mass of the rider and bike, g the acceleration due to Earth's gravity, C_r is the coefficient of rolling resistance that in [40] was found between 0.07 and 1.15, s is the slope or grade, that can be estimated from the barometer (as we did for *walking*), $\rho(T)$ is temperature dependent air density and C_a is the drag coefficient, which can be set to 1.0 $\frac{kg}{m^3}$ and 0.26 respectively according to [40], and A is the frontal drag area, which according to [79] is in average 0.264 m^2.

(a) Side view (b) Top view

Figure 4.13: Forces related to determining the power necessary to move a bike at constant speed V_g given wind vector V_w and road slope s. Image from [40]

Finally, $V_a = V_g + V_w \cos\alpha$ (as shown in Fig. 4.13); where V_w is the wind velocity and V_g the constant bike speed. To avoid having to query a weather web service, V_w is not considered. However, V_g can be calculated by multiplying *distance per revolution* and *pedal frequency per minute*.

Due to a lack of a trustfull ground truth we could not perform any experiment to analyze with how much accuracy a smartphone is able to estimate the *cycling work rate*. Nevertheless, from all of the three parameters necessary to estimate the *work rate* (see Equation 4.4), we consider that the most important or with more weight is the *pedal frequency per minute*, which can be estimated with high accuracy with our step detection algorithm. Furthermore, if we combine the last two parameters from Equation 4.4 we will get the *cycling* speed V_g.

As we can see, V_g is a meaningful parameter for *cycling work rate* estimation. Therefore, by analyzing how well our system estimates V_g, we might predict how accurate the caloric expenditure at cycling will be.

We developed an algorithm to estimate speed and distance from *cycling dataset*. We set a *distance per revolution* = 2.096 meters, which is the wheel circunference of the bike used in *cycling dataset*. Furthermore, due to we are not using the GPS, we can't estimate distance or speed when the bike is moving without user effort (i.e. user is not pedaling). Therefore, heuristically we identified the region of data where the user wasn't pedaling (and consequently when was pedaling as well) (see Fig. 4.14).

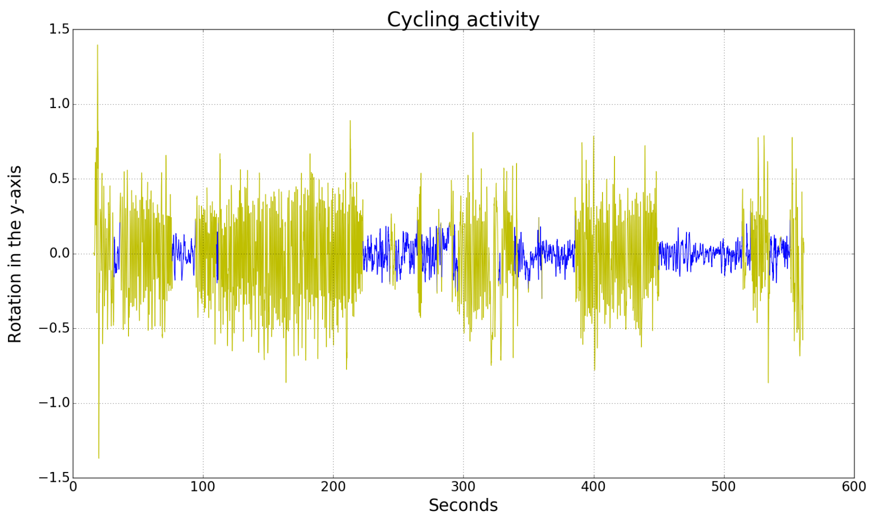


Figure 4.14: Bike pedaling (yellow) and not pedaling (blue) identification.

We applied our step detection algorithm to the segments of data where the user was pedaling to estimate the *pedal frequency* and consequently the *speed* and *distance*. Then, we used the *average speed* from the segments of pedaling and use this value to estimate *distance* in the regions of no pedaling. Therefore, both distances will give us the total distance traveled.

After testing our algoritm on *cycling dataset*, we obtained a accuracy of 0.85±0.11 in the distance estimation (see Table 4.10), an accuracy of 0.51±0.13 in the average speed estimation (see Table 4.11) and 0.68±0.16 in the maximum speed estimation.

Route #	**Real Distance (Km)**	**Estimated Distance (Km)**	**Accuracy**
1	2	2.13	0.935
2	2.5	1.73	0.692
3	1.6	1.62	0.987
4	2	1.6	0.8

Table 4.10: Distance estimations while cycling

Route #	**Real Avg. Speed (Km/h)**	**Estimated Avg. Speed (Km/h)**	**Accuracy**
1	13.8	9.57	0.693
2	16.3	8.71	0.534
3	13.4	6.8	0.507
4	15	4.61	0.266

Table 4.11: Average speed estimations while cycling.

Despite the distance estimation is not required for the ACSM's cycling equation. We consider this is an important metric for the user (i.e. longer distances provide the feeling of more effort). The distance, by using our algorithm, was estimated with a good accuracy. However, the average and maximum speeds estimations had low accuracy values. This is probably due to our algorithm can only estimate with high accuracy the speed when the user is pedaling (due to the algorithm is based on the cycling steps); and in constrast, the bike computer can estimate speed at any time (user pedaling or not pedaling). Furthermore, the maximum speed occurs when the user is not pedaling which consequently affects the average speed measured by the computer.

It is worth to mention that these results do not indicate low accurate energy expenditure estimations. The *work rate* parameter, necessary to estimate calories burned while *cycling*, only considers speeds generated while pedaling. Therefore, as we achieved an accuracy of 91% at detecting *cycling* steps, and in addition the grade (also necessary to calculate *work rate*) can be estimated with 98% of accuracy, we expect *work rate* estimations with accuracies over 90%.

4.7 Cadence and Speed Sensing approach

Another approach that has proved to be useful but less accurate at estimating caloric expenditure at cycling [40] is the cadence and speed sensing approach. Instead of using physics to meausure the work rate and consequently estimate VO_2, the estimation of the oxygen consumption VO_2 in *liters per min* is provided by the following equation:

$$VO_2 = 0.00494(0.261V^3 + 0.67mV)^{0.589}S^{0.168} \tag{4.6}$$

Where V is the bike velocity in (*km/h*) obtained similarly as V_g mentioned in Section 4.6, S is the pedaling rate (in *rpm*) that can be obtained from the step detection algorithm and m that is the total mass of the rider and bike (in kilos) that must be provided by the user. Lastly, we can estimate the calories burned by multiplying VO_2 in $\frac{L}{min}$ by a factor of $5.01\frac{Kcal}{L}$.

Despite being a simpler solution for estimating *cycling* energy expenditure, it has some shortcomings. First, with this method we are underestimating caloric expenditure during uphill trips where the pedaling and bike speed are low, but the effort is greater. Besides, with this method we are not exploiting the embedded barometer sensor capabilities due to we do not need the grade value. However, this approach might be provided as an option to users that are mostly interested in measuring the cadence rather than the final caloric expenditure during bike trips.

4.8 Summary of results

Due to the lack of a trustful ground truth, the performance of the energy expenditure estimation module was evaluated based on the performance of each parameter of the metabolic equations separately. Therefore, if each of these parameters can be estimated with high accuracy then consequently the final caloric estimation will be accurate as well. The parameters to evaluate were the speed, grade, number of steps, step height, and work rate. Results from their estimations are summarized in Table 4.12.

Parameter	**Accuracy**	**Sensor**	**Proxy**	**Activity**
Steps	90%	Gyroscope	Direct	All
Speed	~90%		Steps	*walking* and *jogging*
Grade	98%	Barometer	Direct	*walking* and *jogging*
Step height	74%	Barometer	Direct	*upstairs* and *downstairs*
Work rate	~>90%		Steps and Grade	*cycling*

Table 4.12: ACSM's metabolic equations parameters evaluation.

As we can note steps and grade can be directly estimated with accuracies over 90% by using the gyroscope and barometer sensors. The step height estimation was less accurate (74%). However, this parameter was only estimated to test the granularity level of the barometer. A fixed standard step height value can be used in the metabolic equations instead. The speed and work rate were not directly estimated, but instead based on the results obtained in the steps and grade computations we can expect accuracies over 90% for these two parameters as well. Therefore, according to these results, a highly accurate energy expenditure estimations is possible by using the gyroscope and barometer smartphone's embedded sensors.

Chapter 5

Fitness Mate

In Chapters 3 and 4 there was shown that high accurate values for the activity classification task and caloric expenditure estimation can be achieved with the use of three built-in sensors (accelerometer, gyroscope and barometer) and the ACSM's metabolic equations. However, the activity classification was performed offline by using the Python's linear SVM machine learning classifier, and similarly, the caloric expenditure estimation was performed on different sets of experiments that led us to the conclusion that the caloric expenditure estimation was achievable with high accuracy due to the parameters from the ACSM's equations were estimated with high accuracy as well by using our algorithms (step detection and grade computation).

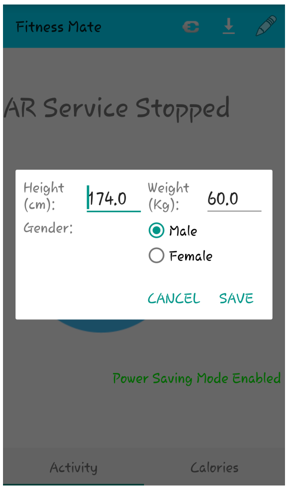
In order to provide a fitness-tracking system we intregated the activity recognition and energy expenditure estimation modules into the *data collection app*, and created *Fitness Mate* [80], which is an Android fitness-tracking mobile application with the features of online automatic activity identification and energy expenditure estimation. Every five seconds the activity perfomed is recognized automatically and the calories expended on that activity are aggregated. The user is able to check at any time of the day the calories burned in total and per activity.

Furthermore, we evaluated the activity recognition module, and performed different experiments to compare our system with two well-known commercial fitness-tracking systems namely *S Health* [81] and *Fitbit* [82].

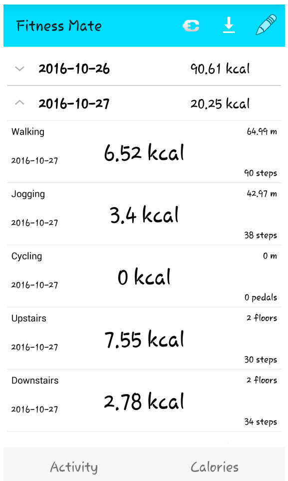
5.1 System overview

Upon installation in the smartphone, *Fitness Mate* will be running as a background service collecting sensing data at 20Hz and recognizing the activity and aggregating its calories consumed every five seconds. In order to preserve smartphone battery life, we also provide a power saving mode, where if *Others* activities are detected continuosly over a two-minutes window, the data collection stops for five minutes and then the activity recognition service restarts.

To perform the online activity classification we are using the Android version of the library for support vector machines Android LIBSVM [83], with the kernel parameter set to "linear" in order to follow the same configuration used in the offline classification. Furthermore, we used the same data collection described in Table 3.2 to build the classification model, which is embedded in the application.



(a) User Profile



(b) Calories Expended

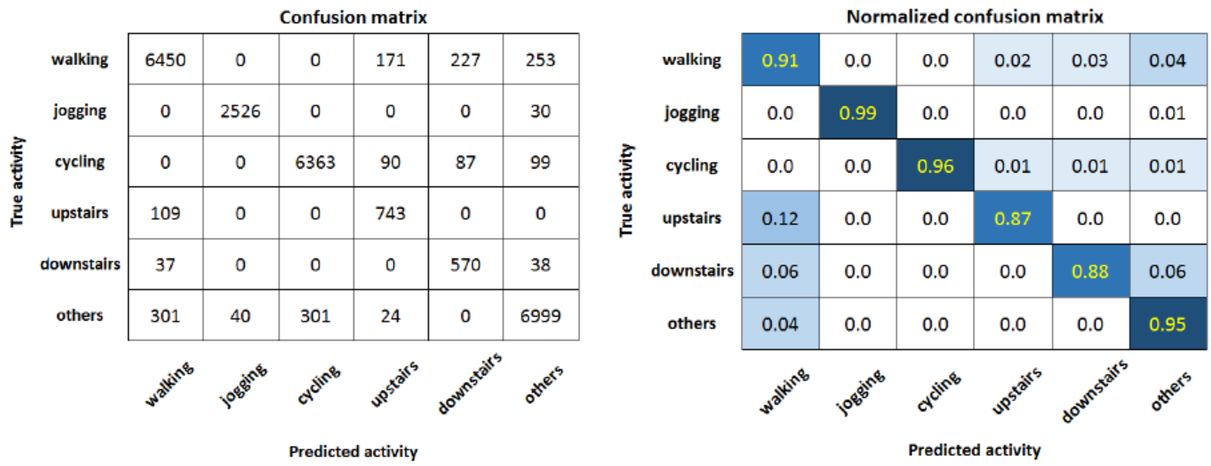
Figure 5.1: Fitness Mate Overview

The user is allowed to provide information related to his height, weigth, and gender (see Fig. 5.1a) to increase the accuracy of the calories estimation. The calories consumed are detailed by activity and in total per day, and can be checked by the user at any time. Besides calories expended, information about the number of steps, distance traveled, and floors climbed is also provided by the system to increase user's motivation (see Fig. 5.1b).

Finally, the application comes with an export button, where the collected data from accelerometer, gyroscope and barometer sensors is available to the user either for research purposes or further analysis.

5.2 Online activity recognition evaluation

To test the accuracy of the embedded Linear SVM activity classifier, the user was asked to perform the *walking*, *jogging*, *cycling*, *upstairs*, and *downstairs* activities for approximately 30 minutes, with the smartphone placed in his pant's front left pocket (armband in case of jogging) and the *Fitness Mate* app running on the same device. By doing so, we obtained the predicted activities dataset. The true activities dataset was obtained from the *Activity labeling app*, similarly as in the data collection described in Section 3.1.



(a) Confusion matrix

(b) Normalized confusion matrix

Figure 5.2: Fitness Mate activity recognition evaluation

From the confusion matrix shown in Fig. 5.2b, we can note that the online classification model achieved with overall 92% accuracy the activity detection task, and 87% in the stairs activities identification. These results are similar to the results obtained with the offline classification described in Section 3.3, and show us that high recognition accuracies can be achieved with the online classification approach.

5.3 System comparison

We performed different experiments to compare our system performance against the *S Health* smartphone application and the *Fitbit* wearable device. We used the latest version of *S Health* at the moment which is version 5.2.1.0001, and the *Fitbit* model used is the *Fitbit Charge HR*. We selected these two systems due to both are good representatives of the two types of fitness-tracking systems namely smartphone-based and wearable-based. Furthermore, both systems have the features of automatic activity inference and caloric expenditure estimation. Nonetheless, it is worth to mention that to the best of our knowledge *S Health* is the only fitness application available in Google Play Store with the automatic activity recognition feature.

The comparisons were based on how accurate all three systems estimated the number of steps for *walking*, *jogging*, *cycling*, *upstairs*, and *downstairs* activities. The ground truth (i.e. real steps) was provided by the step counter button of the *Activity labeling app*. The energy expenditure estimation from each system was also annotated. However, due to a lack of ground truth we could not indicate which system outperforms the others. Nevertheless, we can observe how similar or different their estimations are.

A limitation found either in the *S Health* as in the *Fitbit* system relies in the fact that the activity is only tracked after 10 minutes on the same activity. Therefore, calories expended during small walks around the office, home or any other place as well as any other small period activity will not be counted.

Due to the limitation mentioned above, in order to perform the steps comparison experiments (which consisted on small time periods of activity), the activity to track had to be manually selected in both systems (*S Health* and *Fitbit*).

	Fitness Mate			**S Health**			**Fitbit**		
Real Steps	**Estimated Steps**	**Steps Accuracy**	**Energy Expenditure (kcal)**	**Estimated Steps**	**Steps Accuracy**	**Energy Expenditure (kcal)**	**Estimated Steps**	**Steps Accuracy**	**Energy Expenditure (kcal)**
100	90	0.9000	5.09	101	0.990	6	100	1.000	9
150	136	0.9067	10.0	114	0.760	10	140	0.933	11
250	244	0.9760	17.94	229	0.916	16	163	0.652	12
350	308	0.8800	20.81	354	0.989	23	304	0.869	24
750	692	0.9227	43.51	750	1.000	55	672	0.896	53
Average Steps Accuracy		**0.9171**			**0.931**			**0.870**	

Table 5.1: Walking comparison

	Fitness Mate			**S Health**			**Fitbit**		
Real Steps	**Estimated Steps**	**Steps Accuracy**	**Energy Expenditure (kcal)**	**Estimated Steps**	**Steps Accuracy**	**Energy Expenditure (kcal)**	**Estimated Steps**	**Steps Accuracy**	**Energy Expenditure (kcal)**
100	106	0.9400	10.50	118	0.820	8	131	0.690	6
150	138	0.9200	13.34	166	0.893	12	169	0.873	11
200	196	0.9800	18.56	211	0.945	17	224	0.880	16
250	242	0.9680	22.96	263	0.948	22	252	0.992	19
550	596	0.9164	56.13	575	0.955	49	584	0.938	45
Average Steps Accuracy		**0.9449**			**0.912**			**0.875**	

Table 5.2: Jogging comparison

According to the experiments performed (described in Table 5.1 and Table 5.2), *S Health* and *Fitness Mate* were able to estimate with accuracies over 90% the number steps either in *walking* and *jogging* activities. Having *S Health* a sligthly better performance than *Fitness Mate* at *walking*, but in contrast, *Fitness Mate* was slightly better than *S Health* at *jogging*. In both activities, *Fitbit* achieved an accuracy of 87% at detecting steps. We can also observe that all-three systems provided similar energy expenditure estimations on both activities.

	Fitness Mate			S Health		
Real Steps	**Estimated Steps**	**Steps Accuracy**	**Energy Expenditure (kcal)**	**Estimated Steps**	**Steps Accuracy**	**Energy Expenditure (kcal)**
150	168	0.8800	28.54	82	0.547	7
225	208	0.9244	35.39	156	0.693	11
250	236	0.9440	40.70	174	0.696	14
300	304	0.9867	52.33	190	0.633	15
325	324	0.9969	54.72	221	0.680	17
550	532	0.9673	90.61	425	0.773	30
Average Steps Accuracy		**0.9499**			**0.670**	

Table 5.3: Cycling comparison

Despite *Fitbit* is able to recognize *cycling* activity after 10 minutes riding a bike, it does not allow for manually start tracking this activity. Moreover, due to the sensing device is carried on the wrist, the steps at *cycling* (i.e. bike pedaling) can not be detected. Therefore, this system was not used in the steps comparison experiments at *cycling*.

As described in Table 5.3, *S Health* showed a bad performance at estimating steps at cycling by only achieving an average accuracy of 67% on step detection. In contrast, *Fitness Mate* demonstrated to be capable of estimating with 95% of accuracy the number of steps at cycling.

Something interesting to point out regards to the big difference between the energy expenditure estimations provided by both systems. Due to the lack of a trustful ground truth we can not state which system is closer to the real caloric expenditure value. However, according to some online calories burned calculators [84] [85], it is more likely that our estimations are more accurate than the estimations provided by *S Health*.

	Fitness Mate		
Real Steps	**Estimated Steps**	**Steps Accuracy**	**Energy Expenditure (kcal)**
25	20	0.8000	4.01
37	36	0.9730	8.98
41	32	0.7805	8.03
48	44	0.9167	11.09
61	48	0.7869	12.05
Average Steps Accuracy		**0.8514**	

Table 5.4: Upstairs comparison

	Fitness Mate		
Real Steps	**Estimated Steps**	**Steps Accuracy**	**Energy Expenditure (kcal)**
25	22	0.8800	1.84
37	38	0.9730	3.06
41	44	0.9268	3.58
48	54	0.8750	4.38
61	68	0.8852	5.56
Average Steps Accuracy		**0.9080**	

Table 5.5: Downstairs comparison

Due to neither *S Health* nor *Fitbit* are capable of identifying or tracking stairs activities, the steps evaluation on *upstairs* and *downstairs* was only performed by using our system as shown in Table 5.4 and Table 5.5. The step detection on *upstairs* can be estimated with 85.14% of accuracy and in *downstairs* the estimation is slightly higher with 90.8% of accuracy. It is worth to mention that the accuracy below 90% obtained on *upstairs* experiments is mostly due to misclassifications with walking activity. For example in the last experiment performed on *upstairs*, the total estimated steps were 60 (which is close to the real 61 steps manually counted). Nonetheless, 12 steps were aggregated to the *walking* activity and 48 to the *upstairs* activity.

Results obtained with our system were expected due to the accurate estimations also achieved in Section 4.3. Furthermore, our system has demonstrated to be competitive with well-known fitness-tracking systems achieving accuracies over 90% on the steps estimation at *walking*, *jogging*, *cycling*, and *downstairs*, and 85% while walking upstairs.

Chapter 6

Discussion

Most of current fitness-tracking apps either Android or IOS require constant manual input from the user. Which, in the long term, generates discouragement towards the fitness system. The activity recognition feature implemented on this type of applications reduces considerably the manual input required to track user's activities and consequently increases user's motivation to keep fit. Therefore, in Chapter 3 we implemented an activity recognition module by only using the smartphone as sensing device. The overall accuracy of 91% at detecting activities, shows a competitive system when compared with previous activity recognition systems described in the literature. However, it is important to mention that in contrast with most of the previous works, our classifier only requires basic time-domain features from the accelerometer and gyroscope readings to achieve this result. Additionally, our sensing data was collected under real world conditions, which increases the difficulty at identifying activities.

The stairs activities were the hardest to identify (81% of accuracy), by being sometimes mislabeled as *walking*. Besides the movement similarity between these activities, other factors such as the small number of *upstairs* and *downstairs* collected data (when compared with *walking*) used to train the classification model, and the possible mislabeling between *walking* and the stairs activities during the data collection process (due to small walking periods when going from one stair to the next one), might have contributed to this misclassification. A possible solution to minimize this mislabeling might come from the barometer sensor. For instance, if over a 5 seconds window the activity detected is either *walking*, *upstairs* or *downstairs*, the barometric pressure readings from that window can be used to extract a difference in pressure feature (i.e. difference between the last and first barometric pressure on that window). This feature used as input for another classification model could increase the accuracy at detecting stairs activities.

Besides enhancing the fitness app usability with the addition of an automatic actitivy recognition module, the quality of the activity-related feedback a system can provide to users is important to encourage their commitment to exercise regularly. Previous authors proposed systems able to provide the caloric expenditure while performing activities. Nonetheless, these systems neither reflect the user's effort on their caloric estimations nor exploit the smartphone capabilities to calculate the calories burned. In chapter 4, we implemented an energy expenditure estimator module that uses the ACSM's equations (which allow us to quantify user's effort) and smartphone's built-in sensors, to provide this energy expended feedback to users. With the gyroscope and barometer sensors we achieved accuracies over 90% on the step detection and grade computation experiments. These sensors showed to be useful to provide accurate approximations of the parameters necessary on the ACSM's metabolic equations in order to estimate VO_2 and consequently the energy expenditure. However, these formulas have some limitations based on specific ranges of *walking* and *jogging* speed, *upstairs* and *downstairs* stepping rate and *cycling* work rate, that could decrease the caloric expenditure estimation accuracy.

In Chapter 5 we detailed the integration of the activity recognition and energy expenditure modules into an Android app (Fitness Mate), and compared our system against *S Health* and *Fitbit* fitness-tracking systems. Besides being able to recognize a wider range of activities, our system shows a similar accuracy than *S Health* (over 91%) when counting *walking* and *jogging* steps. Moreover, *Fitness Mate* outperformes *S Health* at detecting *cycling* "steps", where we achieved an accuracy over 94% and *S Health* less than 70%. However, despite the good results obtained from these comparisons, our approach presents several limitations.

Since our classification model was only trained from data collected with the smartphone placed on the pant's front left pocket in a specific position, our system only works under this setting. Furthermore, either the data collected to train the activity classification model and the data collected for the steps and grade computation experiments comes from a single user. Therefore, *Fitness Mate* could be less accurate when tested on more people. Additionally, even if users follow the location-position settings, performing the activities at different paces could affect the activity recognition outcome as well. For instance, *fast walking* could be misclassified as *downstairs* due to the model was trained mostly from activities at a normal pace.

Lastly, despite using energy efficient approaches such as computational inexpensive features, low-power sensors, and online activity classification to avoid constant wireless communication with an external server, the high energy consumption is still an inconvenience. The sensors sampling rate and the background service running all day long will drain the smartphone's battery at a faster rate. Therefore, to solve this issue we implemented a power saving mode, where the application stops collecting sensing information during five minutes in case that only *others* activity has been detected in a two-minutes window. This solution reduces considerably the energy consumption but might affect not counting steps on small activity periods where the application is not listening. The use of single low-power sensor such as the barometer, providing sensing information at a slower rate could be useful to keep a low energy consumption and detecting small periods of activity. The barometer has proved to be able to distinguish between idle and walking [43]. Hence, it could be used to trigger a "wake up" alarm to start the listening event on the other sensors in case a movement is detected.

Chapter 7

Conclusions and Future Work

This research has demonstrated that with the accelerometer, gyroscope, and barometer smartphone's built-in sensors we can identify with high accuracy common daily activities performed while carrying the smartphone in the pant's front pocket and provide an accurate estimation of the calories burned during those activities.

We also found that after the implementation of some energy efficient approaches, the smartphone can be turned into an all-day fitness-tracking system such as most of wearable devices, but with the advantage of no need for the acquisition of an additional gadget. Furthermore, our system showed competitive results against well-known fitness-tracking systems such as *S Health* and *Fitbit* at estimating the number of steps on walking and jogging activities. Being even able to outperform *S Health* step estimation while *cycling*, and to detect walking stairs activities not considered in the widely known systems.

These conclusions and discussions from the previous Chapter suggest where future work should be directed. In the next stage of this research, we plan to create a more general activity classification model by including sensing data from participants with different age and gender. Moreover, this sensing data will be collected with the smartphone placed in other typical locations such as pant's back pocket, shirt pocket, and hands. Similarly, recognitions of daily activities at different paces will be explored as well. Finally, implementations towards preserving the smartphone's battery life are imperative. Experiments with lower sampling rates and low power sensors will be conducted to achieve high energy efficient fitness-tracking applications.

Bibliography

[1] World Health Organization. WHO | Obesity and overweight. `http://www.who.int/mediacentre/factsheets/fs311/en/`. [Online; accessed on 7 September 2016].

[2] Australian Institute of Health and Welfare. Overweight and obesity. `http://www.aihw.gov.au/overweight-and-obesity/`. [Online; accessed on 29 September 2016].

[3] Gayan Padmasekara. Fitness apps, a valid alternative to the gym: a pilot study. *Journal of Mobile Technology in Medicine*, 3(1):37–45, 2014.

[4] John P Higgins. Smartphone applications for patients' health and fitness. *The American journal of medicine*, 129(1):11–19, 2016.

[5] Statista. Number of mobile phone users worlwide from 2013 to 2019 (in billions). `http://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/`. [Online; accessed on 7 September 2016].

[6] Matthias Kranz, Andreas MöLler, Nils Hammerla, Stefan Diewald, Thomas PlöTz, Patrick Olivier, and Luis Roalter. The mobile fitness coach: Towards individualized skill assessment using personalized mobile devices. *Pervasive and Mobile Computing*, 9(2):203–215, 2013.

[7] Sunny Consolvo, David W McDonald, Tammy Toscos, Mike Y Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, et al. Activity sensing in the wild: a field trial of ubifit garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1797–1806. ACM, 2008.

[8] Alger K. Wearable technology is revolutionizing fitness. `http://raconteur.net/technology/wearables-are-the-perfect-fit/`. [Online; accessed on 7 September 2016].

[9] Ashton Pfannenstiel and Barbara S Chaparro. An investigation of the usability and desirability of health and fitness-tracking devices. In *International Conference on Human-Computer Interaction*, pages 473–477. Springer, 2015.

[10] Patrick C Shih, Kyungsik Han, Erika Shehan Poole, Mary Beth Rosson, and John M Carroll. Use and adoption challenges of wearable activity trackers. *iConference 2015 Proceedings*, 2015.

[11] Ana Carolina Tomé Klock and Isabela Gasparini. A usability evaluation of fitness-tracking apps for initial users. In *International Conference on Human-Computer Interaction*, pages 457–462. Springer, 2015.

[12] R Seiler and M Hüttermann. E-health, fitness trackers and wearables—Use among Swiss students. In *Advances in Business-Related Scientific Research Conference 2015 Proceedings*, 2015.

[13] Paul Krebs and Dustin T Duncan. Health app use among US mobile phone owners: a national survey. *JMIR mHealth and uHealth*, 3(4), 2015.

[14] Kewei Sha, Guoxing Zhan, Weisong Shi, Mark Lumley, Clairy Wiholm, and Bengt Arnetz. Spa: a smart phone assisted chronic illness self-management system with participatory sensing. In *Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, page 5. ACM, 2008.

[15] Haibo Ye, Tao Gu, Xiaorui Zhu, Jinwei Xu, Xianping Tao, Jian Lu, and Ning Jin. Ftrack: Infrastructure-free floor localization via mobile phone sensing. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pages 2–10. IEEE, 2012.

[16] Andrew Ofstad, Emmett Nicholas, Rick Szcodronski, and Romit Roy Choudhury. Aampl: accelerometer augmented mobile phone localization. In *Proceedings of the*

first ACM international workshop on Mobile entity localization and tracking in GPS-less environments, pages 13–18. ACM, 2008.

[17] Oresti Banos, Claudia Villalonga, Miguel Damas, Peter Gloesekoetter, Hector Pomares, and Ignacio Rojas. Physiodroid: Combining wearable health sensors and mobile devices for a ubiquitous, continuous, and personal monitoring. *The Scientific World Journal*, 2014, 2014.

[18] JBJ Bussmann, WLJ Martens, JHM Tulen, FC Schasfoort, HJG Van Den Berg-Emons, and HJ Stam. Measuring daily behavior using ambulatory accelerometry: the activity monitor. *Behavior Research Methods, Instruments, & Computers*, 33(3):349–356, 2001.

[19] Mitja Luštrek, Božidara Cvetković, and Simon Kozina. Energy expenditure estimation with wearable accelerometers. In *2012 IEEE International Symposium on Circuits and Systems*, pages 5–8. IEEE, 2012.

[20] Seon-Woo Lee and Kenji Mase. Activity and location recognition using wearable sensors. *IEEE pervasive computing*, 1(3):24–32, 2002.

[21] Paul Lukowicz, Holger Junker, Mathias Stäger, Thomas von Bueren, and Gerhard Tröster. Wearnet: A distributed multi-sensor system for context aware wearables. In *International Conference on Ubiquitous Computing*, pages 361–370. Springer, 2002.

[22] Ling Bao and Stephen S Intille. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing*, pages 1–17. Springer, 2004.

[23] Kristof Van Laerhoven, Albrecht Schmidt, and H-W Gellersen. Multi-sensor context aware clothing. In *Wearable Computers, 2002.(ISWC 2002). Proceedings. Sixth International Symposium on*, pages 49–56. IEEE, 2002.

[24] Hikaru Inooka, Yasuaki Ohtaki, Hiroshi Hayasaka, Akihiro Suzuki, and Ryoichi Nagatomi. Development of advanced portable device for daily physical assessment. In *2006 SICE-ICASE International Joint Conference*, pages 5878–5881. IEEE, 2006.

[25] Tamara Denning, Adrienne Andrew, Rohit Chaudhri, Carl Hartung, Jonathan Lester, Gaetano Borriello, and Glen Duncan. BALANCE: towards a usable pervasive wellness application with accurate activity inference. In *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, page 5. ACM, 2009.

[26] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In *International Conference on Pervasive Computing*, pages 1–16. Springer, 2006.

[27] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[28] Edmond Mitchell, David Monaghan, and Noel E O'Connor. Classification of sporting activities using smartphone accelerometers. *Sensors*, 13(4):5317–5337, 2013.

[29] Lin Sun, Daqing Zhang, Bin Li, Bin Guo, and Shijian Li. Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. In *International Conference on Ubiquitous Intelligence and Computing*, pages 548–562. Springer, 2010.

[30] Daniel Kelly and Brian Caulfield. An investigation into non-invasive physical activity recognition using smartphones. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3340–3343. IEEE, 2012.

[31] De Feng Guo, Bin Liu, Xiao Tian Jin, and Hong Jian Liu. Human activity recognition using smart-phone sensors. In *Applied Mechanics and Materials*, volume 571, pages 1019–1029. Trans Tech Publ, 2014.

[32] Wanmin Wu, Sanjoy Dasgupta, Ernesto E Ramirez, Carlyn Peterson, and Gregory J Norman. Classification accuracies of physical activities using smartphone motion sensors. *Journal of medical Internet research*, 14(5):e130, 2012.

[33] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors*, 16(4):426, 2016.

[34] Stefan Dernbach, Barnan Das, Narayanan C Krishnan, Brian L Thomas, and Diane J Cook. Simple and complex activity recognition through smart phones. In *Intelligent*

Environments (IE), 2012 8th International Conference on, pages 214–221. IEEE, 2012.

[35] Yi He and Ye Li. Physical activity recognition utilizing the built-in kinematic sensors of a smartphone. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[36] Muhammad Shoaib, Hans Scholten, and Paul JM Havinga. Towards physical activity recognition using smartphone sensors. In *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, pages 80–87. IEEE, 2013.

[37] Alvina Anjum and Muhammad U Ilyas. Activity recognition using smartphone sensors. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 914–919. IEEE, 2013.

[38] Jilong Liao, Zhibo Wang, Lipeng Wan, Qing Charles Cao, and Hairong Qi. Smart diary: A smartphone-based framework for sensing, inferring, and logging users' daily life. *IEEE Sensors Journal*, 15(5):2761–2773, 2015.

[39] Arindam Ghosh and Giuseppe Riccardi. Recognizing human activities from smartphone sensor signals. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 865–868. ACM, 2014.

[40] Andong Zhan, Marcus Chang, Yin Chen, and Andreas Terzis. Accurate caloric expenditure of bicyclists using cellphones. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 71–84. ACM, 2012.

[41] Kartik Muralidharan, Azeem Javed Khan, Archan Misra, Rajesh Krishna Balan, and Sharad Agarwal. Barometric phone sensors: More hype than hope! In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, page 12. ACM, 2014.

[42] Salvatore Vanini, Francesca Faraci, Alan Ferrari, and Silvia Giordano. Using barometric pressure data to recognize vertical displacement activities on smartphones. *Computer Communications*, 87:37–48, 2016.

[43] Kartik Sankaran, Minhui Zhu, Xiang Fa Guo, Akkihebbal L Ananda, Mun Choon Chan, and Li-Shiuan Peh. Using mobile phone barometer for low-power transportation context detection. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 191–205. ACM, 2014.

[44] Ian Anderson, Julie Maitland, Scott Sherwood, Louise Barkhuus, Matthew Chalmers, Malcolm Hall, Barry Brown, and Henk Muller. Shakra: tracking and sharing daily activity levels with unaugmented mobile phones. *Mobile Networks and Applications*, 12(2-3):185–199, 2007.

[45] Jun Yang. Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pages 1–10. ACM, 2009.

[46] William L Haskell, I-Min Lee, Russell R Pate, Kenneth E Powell, Steven N Blair, Barry A Franklin, Caroline A Macera, Gregory W Heath, Paul D Thompson, and Adrian Bauman. Physical activity and public health: updated recommendation for adults from the american college of sports medicine and the american heart association. *Circulation*, 116(9):1081, 2007.

[47] Luis Miguel Soria Morillo, Luis Gonzalez-Abril, Juan Antonio Ortega Ramirez, and Miguel Angel Alvarez de la Concepcion. Low energy physical activity recognition system on smartphones. *Sensors*, 15(3):5163–5196, 2015.

[48] Henar Martín, Ana M Bernardos, Josué Iglesias, and José R Casar. Activity logging using lightweight classification techniques in mobile devices. *Personal and ubiquitous computing*, 17(4):675–695, 2013.

[49] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International Workshop on Ambient Assisted Living*, pages 216–223. Springer, 2012.

[50] Alvin Prayuda Juniarta Dwiyantoro, I Gde Dharma Nugraha, and Deokjai Choi. A simple hierarchical activity recognition system using a gravity sensor and accelerometer on a smartphone. *International Journal of Technology*, 7(5):831, 2016.

[51] Chung-Tse Liu and Chia-Tai Chan. A fuzzy logic prompting mechanism based on pattern recognition and accumulated activity effective index using a smartphone embedded sensor. *Sensors*, 16(8):1322, 2016.

[52] Scott E Crouter, Kurt G Clowers, and David R Bassett. A novel method for using accelerometer data to predict energy expenditure. *Journal of applied physiology*, 100(4):1324–1331, 2006.

[53] Daniel P Heil. Predicting activity energy expenditure using the actical® activity monitor. *Research quarterly for exercise and sport*, 77(1):64–80, 2006.

[54] C Bouten, K Westerterp, Maarten Verduin, and J Janssen. Assessment of energy expenditure for physical activity using a triaxial accelerometer. *Medicine and science in sports and exercise*, 23(1):21–27, 1994.

[55] Jonathan Lester, Carl Hartung, Laura Pina, Ryan Libby, Gaetano Borriello, and Glen Duncan. Validated caloric expenditure estimation using a single body-worn sensor. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 225–234. ACM, 2009.

[56] Android Developers. Sensors overview. `https://developer.android.com/guide/topics/sensors/sensors_overview.html`. [Online; accessed on 28 September 2016].

[57] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. Activity recognition from accelerometer data. In *AAAI*, volume 5, pages 1541–1546, 2005.

[58] K Aminian, Ph Robert, EE Buchser, B Rutschmann, D Hayoz, and M Depairon. Physical activity monitoring based on accelerometry: validation and comparison with video observation. *Medical & biological engineering & computing*, 37(3):304–308, 1999.

[59] F Foerster, M Smeja, and J Fahrenberg. Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Computers in Human Behavior*, 15(5):571–583, 1999.

[60] Kamiar Aminian, Philippe Robert, E Jequier, and Yves Schutz. Estimation of speed and incline of walking using neural network. *IEEE Transactions on Instrumentation and Measurement*, 44(3):743–746, 1995.

[61] RENÉ Herren, ANDREA Sparti, KAMIAR Aminian, and YVES Schutz. The prediction of speed and incline in outdoor running in humans using accelerometry. *Medicine and science in sports and exercise*, 31(7):1053–1059, 1999.

[62] Weka. Data mining software. `http://http://www.cs.waikato.ac.nz/ml/weka/`. [Online; accessed on 26 October 2016].

[63] Scikit learn. Machine learning in python. `http://scikit-learn.org/stable/`. [Online; accessed on 26 October 2016].

[64] Scikit-learn. sklearn.svm.svc — scikit-learn 0.18 documentation. `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC`. [Online; accessed on 10 September 2016].

[65] Scikit-learn. sklearn.tree.decisiontreeclassifier — scikit-learn 0.18 documentation. `http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier`. [Online; accessed on 10 September 2016].

[66] Paul Webb, James F Annis, and SJ Troutman. Energy balance in man measured by direct and indirect calorimetry. *The American journal of clinical nutrition*, 33(6):1287–1298, 1980.

[67] J Speakman. *Doubly labelled water: theory and practice*. Springer Science & Business Media, 1997.

[68] James A Levine. Measurement of energy expenditure. *Public health nutrition*, 8(7a):1123–1132, 2005.

[69] S Gibson and A Numa. The importance of metabolic rate and the folly of body surface area calculations. *Anaesthesia*, 58(1):50–55, 2003.

[70] AC Guyton and JE Hall. Textbook of medical physiology 8th edn (philadelphia, pa: Saunders). 1991.

[71] American College of Sports Medicine et al. *ACSM's guidelines for exercise testing and prescription*. Lippincott Williams & Wilkins, 2013.

[72] Suaoki. Suaoki wireless bike cadence sensor. `http://www.suaoki.com/product-g_29.html`. [Online; accessed on 15 October 2016].

[73] SH Shin, MS Lee, CG Park, and Hyun Su Hong. Pedestrian dead reckoning system with phone location awareness algorithm. In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pages 97–101. IEEE, 2010.

[74] Ying-Wen Bai, Chia-Hao Yu, and Siao-Cian Wu. Using a three-axis accelerometer and GPS module in a smart phone to measure walking steps and distance. In *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*, pages 1–6. IEEE, 2014.

[75] MathWorks. Measure rate of rotation around x, y, and z axes. `http://au.mathworks.com/help/supportpkg/android/ref/gyroscope.html?requestedDomain=www.mathworks.com`. [Online; accessed on 9 September 2016].

[76] Azkario Rizky Pratama, Risanuri Hidayat, et al. Smartphone-based pedestrian dead reckoning as an indoor positioning system. In *System Engineering and Technology (ICSET), 2012 International Conference on*, pages 1–6. IEEE, 2012.

[77] Mayu Sumida, Teruhiro Mizumoto, and Keiichi Yasumoto. Estimating heart rate variation during walking with smartphone. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 245–254. ACM, 2013.

[78] Geoscience Australia 2. ELVIS (Elevation Information System). `http://www.ga.gov.au/elvis/`. [Online; accessed on 1 September 2016].

[79] James C Martin, Douglas L Milliken, John E Cobb, Kevin L McFadden, and Andrew R Coggan. Validation of a mathematical model for road cycling power. *Journal of applied biomechanics*, 14:276–291, 1998.

[80] Nestor Cabello. Fitness mate – fitness-tracking application. https://play.google.com/store/apps/details?id=unimelb.steven.fitnessapp. [Online; accessed on 25 October 2016].

[81] S health. S health | Start a Health Challenge. http://shealth.samsung.com/. [Online; accessed on 25 October 2016].

[82] Fitbit. Fitbit official site for activity trackers & more. https://www.fitbit.com/au. [Online; accessed on 25 October 2016].

[83] Yu-Chih Tung. Android libsvm. https://github.com/yctung/AndroidLibSvm. [Online; accessed on 25 October 2016].

[84] Exploratorium: the museum of science, art and human perception. Science of cycling: Aerodynamics & wind resistance | exploratorium. https://www.exploratorium.edu/cycling/aerodynamics1.html. [Online; accessed on 26 October 2016].

[85] Mapmyride. Caloric calculator. http://www.mapmyride.com/improve/calorie_calculator/. [Online; accessed on 26 October 2016].