A WEARABLE DATA COLLECTION PLATFORM WITH SMART ANNOTATION CAPABILITIES

A Thesis

by

ROGER FERNANDO SOLIS CASTILLA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Bobak J. Mortazavi
Co-Chair of Committee,	Roozbeh Jafari
Committee Member,	Xia Hu
Head of Department,	Dilma Da Silva

August 2019

Major Subject: Computer Science

ABSTRACT

Remotely tracking users' activity and physiology can help on disease treatment and health monitoring. For example, in nutrition management, tracking food intake helps on weight control. However, to train tracking algorithms, annotated data is needed which is typically obtained manually. Users' manual annotation is challenging as it's affected by factors such as recall bias and may become a burden, causing them to stop annotating. Automatic approaches exist, but they may not personalize to individual users, resulting in inaccurate annotations. Therefore, personal pattern identification and adaptation are needed to achieve a satisfactory annotation process. We present a system capable of personal patterns' identification and intelligent data annotation for accurate personal monitoring without burdening the user.

DEDICATION

To my mother and father, my family, and my beloved wife. Because without their unconditional love and support this wouldn't been possible.

ACKNOWLEDGMENTS

I would like to thank my family who was always there unconditionally motivating me for finishing the degree, in particular my parents Aurora Elizabeth Castilla Gongora and Alfredo Jose Solis Ojeda whom never stopped believing in me. I would also like to thank my now wife and best friend Elisa Alejandra Cordova Zapata, who was always by my side regardless of the time and distance. Also, thanks to all my friends that were always there cheering me up when I needed it the most.

I would like to extend my special thanks to my committee chair and advisor Dr. Bobak J. Mortazavi who guided me to accomplish this research project through his advice. Also thanks to my co-chair Dr. Roozbeh Jafari who was always willing to collaborate with me.

Finally, I would like to extend my gratitude to the institutions that financially supported me throughout my studies to make this degree possible. Thanks to CANIETI and CONCITEY for giving me the opportunity to get to know Texas A&M University through a summer internship program. Also, thanks to CONACYT and SIIES that financially supported my first two yeast of study through a scholarship, and thanks to the TEES for financially supporting my last year through a research assistantship with Dr. Mortazavi.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professors Bobak J. Mortazavi, and Xia Hu of the Department of Computer Science and Engineering, and Professor Roozbeh Jafari of the Departments of Biomedical Engineering, Computer Science and Engineering, and Electrical and Computer Engineering.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported in part by a scholarship from two Mexican institutions CANIETI and CONACYT, and by a graduate assistantship from Texas A&M University.

NOMENCLATURE

BLE	Bluetooth Low Energy
IMU	Inertial Measurement Unit
APP	Application
GPS	Global Positioning System
API	Application Programming Interface
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory networks

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Personal health monitoring and tracking has become increasingly feasible through pervasive and ubiquitous wearable sensors such as smartwatches [1]. Systems built around these devices are able to track personal activity and query individuals for behavior and health [2, 3], and are better at tracking than smartphones that are often in proximity but not on the individual [4]. In order to successfully track users, tracking algorithms are trained on datasets that are often obtained from groups of users. However, the algorithms might be used in different groups of users. As a consequence, it is necessary to personalize the recognition algorithms to each user in order to accomplish successful personal monitoring. Additionally, large amounts of annotated data are needed in order to train such tracking algorithms.

The annotation process, is a process on which labels for data used for training recognition algorithms is obtained. This process is usually performed in laboratories [5] which can be considered as controlled environments. When users go to laboratories for data collection, the researcher is usually who annotates the data, and he does so following protocol which dictates how the data should be collected. As a consequence, the collected data does not necessarily represent that of natural environments. Therefore, any algorithm trained in these datasets tend to perform poorly when tested in natural environments [6].

Different works have reported this type of data collection issues. The reliability of accelerometry for activity recognition in and out the laboratory has been investigated [6]. The authors found out that despite the miss-classification rates for activity recognition with data collected in-lab was as low as 9%, the miss-classification rates increased significantly in the out-lab data up to 33.3%. This work supports the belief that the laboratory settings can bias data and hide valuable information resulting in algorithms performing poorly in natural environments. Different classifiers for the activity recognition task [5] have also been investigated. In this work, the authors have mentioned the drawbacks of data collection under controlled environments, which they argued was diminished by following a semi-naturalistic protocol. This semi-naturalistic protocol has as a main goal to ask the participants what to do in an "open" way without giving out specific details on how a given activity should be performed. Also, no supervision was made once the experiments began. Their idea was to not bias the data collection by telling the user how to perform the activities. Despite their approach seems more realistic, it still lacks of flexibility. The subjects were recruited at a laboratory and the activities were all performed following a circuit designed by the researchers. As a result, there were no external factors that might be present in a natural environment and that can affect the way the subjects perform the given activities. A solution to this problem of bias, is to train the algorithms on annotated data obtained in natural environments. However, obtaining annotated data in natural environments is challenging as it usually implies multiple user's interactions.

In natural environments, annotations are usually performed manually by users. This approach is often affected by user's recall bias and if the process is not carried carefully, it can become a burden to users who then stop the data annotation process, resulting in a lack of annotations. Manual logging of activities such as dietary intake, activities performed, and locations visited, is a burdensome process that often sees greatly reduced user compliance [7, 8]. An example of this reduced user's compliance can be seen in remote cardiac rehabilitation. Remote cardiac rehabilitation allows patients to perform their rehabilitation exercises remotely without having to be physically at the clinic facilities [9]. To accomplish this monitoring, patients are asked to self-log their exercise and activities and communicate their notes to their physicians. Unfortunately, this often becomes a burden, which causes them to drop out the rehabilitation programs [10, 11]. Similarly, a study in mental wellness reported reduced adherence rates on their users when they were asked to answer an average of 8 surveys a day [8]. On the other hand, in dietary monitoring, adherence reduction has also been observed, as the participants leave what is known as the novelty period which is a period of motivation just because the novelty of the tool [12]. People forget to log food intake, do not want to log for privacy reasons, or simply find the benefits are not outweighed by the burden of annotating their eating patterns. If systems leverage technology that involves additional passive sensing, observations captured can infer logging and reduce the burden of manual logging by automatizing most of the process, likely increasing user adherence. In other words, automatic approaches can improve the annotation process through reduced annotation burden.

Different approaches to automatic data annotation have been proposed that aim to ask users for annotations only when is needed. Such algorithms are mostly based on the classification model's probability output. For example, semi-supervised approaches, such as self-learning [13], and co-training [14], use the probability returned by the classifier to guide their labeling process, requiring annotations only when the model's output probability is below a threshold. However, these approaches tend to overfit and may ignore samples for which the classifier's probability output was high, even though the assigned label was incorrect. Automatic labeling approaches, such as active learning (AL) and experience sampling methods (ESM) that obtain labels directly from the user increase accuracy at higher rates [15, 16]; however, such approaches do not limit the number of interactions, likely impacting the adherence [17]. We need a system for data collection and annotation that is pervasive enough to be used for long periods of time, capable of collecting data in natural environments, and that has smart annotation capabilities to reduce user's annotation burden.

Smartwatches are a flexible platform that provide a potential solution to optimizing this manual logging burden. Studies have shown that recognition of activities can be performed through the analysis of sensor data captured by smartwatches. For example, accelerometer and gyroscope data captured from the watch can be used for the recognition of several activities, such as walking or jogging [18], for the detection of eating moments [17], and for posture detection [19]. Therefore, we propose a smartwatch based system for automatic data annotation that intelligently prompts user for annotations only when required with aims at minimizing user's annotation burden. In this thesis, we introduce a smartwatch based platform for longitudinal data collection. We also provide an offline analysis for intelligent annotation and personalization. Additionally, we include an online analysis on the impact of running deep learning models on a device with limited computing power such as a smartwatch. The analysis provided by this work are all performed on real world data obtained through a diet monitoring study.

The remaining chapters discuss the related work, approach, and the results of this thesis. Chapter 2 discusses the relevant work related to data collection platforms, annotations in natural environments, and diet monitoring. The goals of this thesis are listed and discussed in Chapter 3. The approach taken in this thesis is presented in Chapter 4 and the results obtained are discussed in Chapter 5. Finally, the conclusions and future directions of this work are discussed in Chapter 6.

2. RELATED WORK

Data collection platforms are often developed by scientists in order to collect the data on which they want to perform their research. However, they often do not care about device's pervasiveness as well as annotation burdens as the data collection platforms are usually used in laboratory settings. Automatic annotation approaches have also been proposed that aim at minimizing user's imposed burden on manual annotations. They are mostly based on the prediction model's output probability. On the other hand, annotations in diet monitoring are important as they give a better understanding of user's patterns of behavior. In this chapter, we discuss some related works on data collection platforms, automatic annotation approaches, and diet monitoring.

2.1 Data collection platforms

Data collection platforms are not new and different works have tried to provide a tool for data collection that facilitates the data collection process. A data collection platform for mental health research which captures information provided by the user has been proposed [20]. In this platform, data is manually introduced by the user, and data such as HR and activity information are also captured. Despite they mention that the platform is able to collect data from an electrocardiogram (ECG) sensor, they didn't mention if their app supports the addition of any other external sensor besides the ECG. Despite this is a good way to obtain information from the user, the burden that this implies is not desired. Another data collection platform called OmniTrack [21] is a smartphone based, semi-automated tracking app that attempts to overcome the main shortcomings of already available self-tracking apps. OmniTrack allowed subjects to track several custom activities such as sleeping, or exercising. It also allowed users to log their tracking information either manually, or semi-automatically through the configuration of triggers that automatically log information based on events. However, even if the users decide to set up triggers that would help them to log their information through reminders, the system did not learn the user's behavior and therefore its still highly dependent on the users. The result is a constant need of tuning the triggers to adapt to users.

On the other hand, in a longitudinal study a smartwatch and a smartphone were used as a data collection platform for personal risk detection [22]. In their study, they collected data 24 hours a day for 1 week from multiple sensors found in a Microsoft band. During the data collection process, the participants manually logged their activities but longer term data collection was not investigated.

Another approaches to data collection platforms propose both, the hardware that collects the data and the software that handles the collected information. A data collection system using a smartwatch, smartphone, and microlocation information provided by Bluetooth low energy (BLE) beacons has been introduced [23]. Despite they claim that their proposed framework can be configured to integrate more sensors, their work is based on custom-made beacons which is a not so easily scalable solution. Also, they do not cover a personalization of the models and their platform design might not be pervasive enough as it would require users not only to carry a smartwatch, but also a smartphone as depicted in Figure 2.1. Another data collection platform has also been proposed motivated by the idea of having a centralized device for activity related data collection instead of multiple devices [24]. The platform is a wearable device capable of performing activity recognition based on multiple sensor data. Despite the platform is able to wireless transfer the data to a mobile device, the data that can be collected is restricted by the type of sensors that have been embedded into their hardware, which are sensors they have determined that are required to properly recognize subjects' activities. In other words, this device has been built with the solely purpose of collecting activity related data and does not allow extra sensors to be added, at least not in an easy way.

2.2 Annotations

The influence of the laboratory setup on the data collection process has previously been observed on studies on which the performance of predictive models decreased significantly when comparing test results in-the-lab against out-the-lab settings [6]. It has also been suggested that allowing the users to collect and label their own data with no influence by researchers for the activity prediction task can improve their models accuracy [5]. The authors argue that most data is

Figure 2.1: Data collection platform with BLE for micro-location. Reprinted from: [23]

collected on laboratory setups under controlled situations, therefore, the models built out of such data does not generalize well and when deployed in a real word setting the performance drops significantly. Having users collect and label their own data allowed them to show a good predictive performance in a semi-real world setting. However, they did not propose a remote data collection platform and their results were based on data collected from their experiments performed at their laboratory, which was just a simulation of a real world setting. Therefore, its performance in a real-world setup it's yet to be analyzed.

Different methods have been proposed to annotate data in natural environments that try to overcome the problem of annotations on a laboratory. Some of them involve automatic approaches and some others require the user to manually provide labels on the data. Despite manual approaches exist, they often require significant interaction from the user, this causes them to stop using the system in the long term since the compliance levels are affected by the burdensome task of annotating data too often [8, 25]. This burden was reported in mental wellness study where a platform for the assessment of mental stress in college students was proposed [8]. The platform automatically collected activity data from a smartphone and asked an average of 8 questionnaires a day, which ranged from 3 to 13 questions each. They observed that on average, 6 questionnaires were answered a day during the first days of the study and the number of solved surveys diminished as the study progressed. By the end of the study, the number of responses significantly dropped to 2 surveys a day. In dietary monitoring, adherence reduction has also been observed, as the participants leave the novelty period [12]. People forget to log food intake, do not want to log for privacy reasons, or simply find the benefits are not outweighed by the burden. If systems leverage technology that involves additional passive sensing, observations captured can infer logging and reduce the burden of manual logging by automatizing most of the process, likely increasing user adherence.

Some other approaches to annotations attempt to reduce the user burden by asking the users for annotations only when needed instead of asking indiscriminately, and sometimes not asking them at all. To this end, a base classifier is trained on the data from which annotations will be taken. The base classifier is then trained using a small dataset with few available labels and the classifier is given to the user for its use. Self-learning approaches use the probability output of the classifier when this assigns labels to samples [13]. When predictions are made, the sample used for prediction is annotated with its predicted label if the probability returned by the classifier is above a threshold. This prediction is considered as a valid annotation, and therefore, its added to the training set for the retraining of the classifier. Despite this approach improves the accuracy of the model without having to interact with the user, it can add noise to the dataset when high confidence predictions are made, but they are wrongly labeled.

Some other approaches have tried to improve the simplistic self-learning approach by adding multiple classifiers. Instead of having a single classifier predicting the label for a sample, approaches such as co-training [14] , En-Co-Training [26], and democratic co-learning [27] use multiple classifiers to predict a label for each sample. Co-training uses two classifiers of the same type (e.g. Decision trees) each one trained from two different and independent views of the data. When each classifier issues a prediction, the label is defined as the annotation for the sample if the two classifiers agree on the same prediction and each classifier augments the training set of the other and a retrain process is made to improve the classifier [14]. However, the restriction on the two views of the data on which its independence has to be ensured prevents this approach from being applicable to multiple problems.

En-Co-Training tries to improve the co-training approach by integrating an additional classifier.

However, the restriction on which different views are required from the data imposed by co-training is removed, and now, three classifiers from different types are trained on the same view of the data. When the classifiers predict a label for a sample, the new sample and label are added to the training set if the three classifiers agree on the same label [26]. On the other hand, democratic co-learning improves co-training in a similar way as en-co-training. The only difference is that now each classifier has its own training set. When a prediction is made, the sample is annotated with the label agreed by the majority of the classifiers. Then, the sample and label are added to each the training set of each classifier that disagree with the majority [27].

Despite the automatic approaches as described before are desirable since they do not have interactions with the users, methods on which interaction from the user is needed have shown better accuracy improvements on the base classifier when annotations are made [16]. However, one of the key points on this approaches is to obtain the information from the user without representing a burden. Previous studies on ESM have reported that when the interaction from the user is too often, the interest from them on providing information is reduced, causing a loss of adherence [28]. Additionally, cumbersome approaches for requesting labels to the user can result in annotation problems such as annotating data the wrong way, or not annotating it at all [25]. Therefore, different approaches attempt to reduce the number of interactions needed from the user with more intelligent algorithms for requesting annotations.

Similarly to self-learning, active learning approaches based on the uncertainty of a classifier have been proposed. However, as a difference from the self-learning methods, this approach asks for an annotation when the uncertainty levels are below a threshold [29]. It has also been reported that the interaction between the user and the system has to be minimized in order to ensure adherence, yet, obtaining important labels is also desirable since from that depends a good re-training process. However, most of the approaches that seek to ask for annotations only when the uncertainty level is low, do not pay special attention on limiting the number of annotations requested a day [30]. In other words, despite they claim the interactions are reduced by setting up a proper threshold, there is no restriction in the number of annotations required by the user.

2.3 Diet monitoring

Several approaches have been developed for the prediction of eating moments. Audio processing, inertial measurement unit (IMU) data processing, and the design of wearable devices were among the most popular solutions proposed by researchers. For example, the detection of family eating moments was designed through the analysis of audio recordings taken from smartphones placed at the family's house [31]. With the processing of audio signals, the researchers were able to identify gathering sounds, which were then translated into family eating moments. Additionally, when available, they used motion data captured from smartwatches worn by some members of a family for their predictions. However, their goal was not to detect eating moments of particular subjects when alone. Furthermore, their approach was only tested at in-home settings, which means that meal predictions can be only made at their home and not a restaurant, for example.

Another approach to the recognition of eating moments involved the usage of IMU data only from a smartwatch [17]. In order to train their classifiers for the recognition of eating moments, authors had to collect labeled data by capturing a video of each participant in their lab settings. Then, they tested their platform in natural settings, where they asked subjects to hang a smartphone around their necks, which captured images every 60 seconds. This way they were able to manually label the starting and ending times of the eating moments. The smartphone was also used as a hub for the data captured by the smartwatch. Regardless of the effectiveness of their labeling approach, it involved a large number of interactions from the users on the processing of the captured images as they needed to look at the images and make sure they were correct as depicted in Figure 2.2 In fact, the authors mentioned that in an attempt to run a longitudinal trial the participant had to carry a power bank, which was reported as burdensome by the participant.

Other approaches to eating detection developed wearable devices that are placed around the neck or close to the mouth for detecting chewing events like the EarBit [32]. The EarBit is a wearable device composed of two IMU units, one proximity sensor, and one microphone. Through the analysis of the signals collected by their sensors, authors were able to successfully predict eating moments. Furthermore, in order to achieve labeling in their proposed solution, video recordings

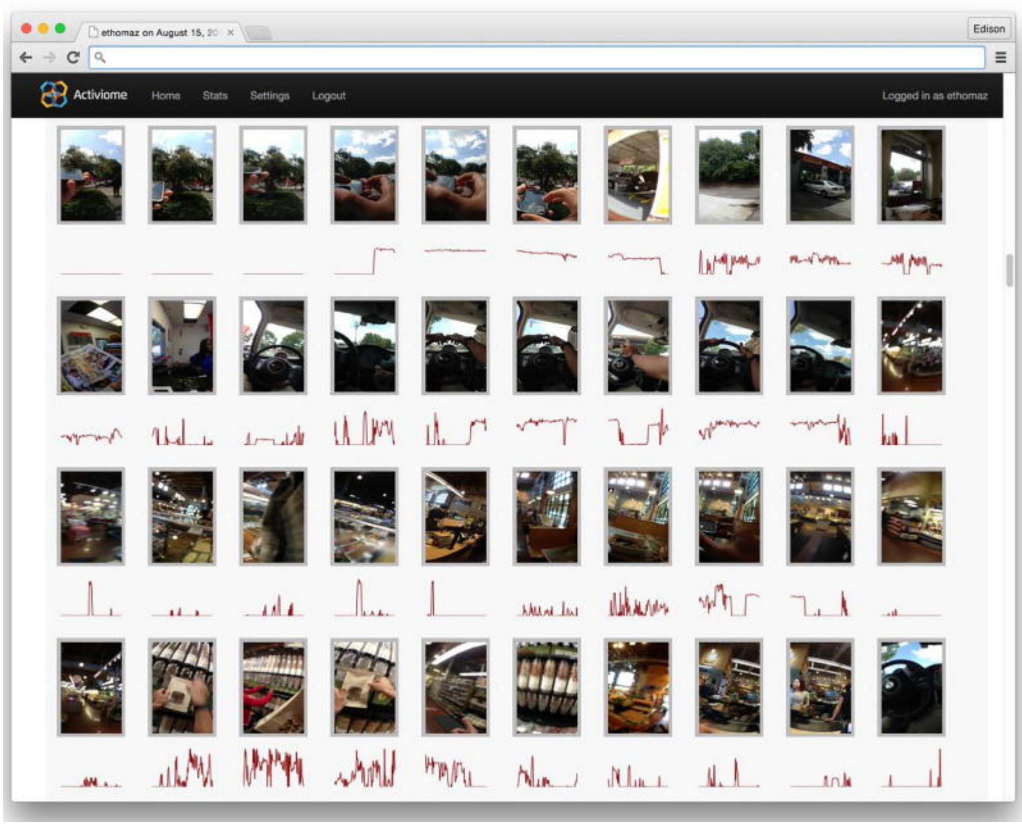

Figure 2.2: Annotation process of images taken in natural environments. Reprinted from: [17]

were needed, which might require users or researchers to manually process such information. A similar work built a wearable device composed of a necklace with an embedded piezoelectric sensor that captured vibrations in the neck of the users [33]. Whenever the users would eat, the sensor would capture the vibrations produced by muscle contractions in the throat. Such vibrations would then be correlated to the eating movements from which they were able to recognize solids and liquid food swallow events. Similarly, another approach used piezoelectric and accelerometer sensors embedded in eyeglasses for the real-time recognition of eating movements and non-eating moments [34]. The development of new wearable sensors should be investigated further for usability and pervasiveness. Smartwatches are more ubiquitous devices that are more likely to be worn long-term.

3. RESEARCH GOALS

After discussing the motivation for this thesis in Chapter 1 and some related works in Chapter 2, this chapter describes the specific goals pursued on this work. In order to provide a pervasive data collection platform with smart annotation capabilities, the main goals were proposed. The following sections describe each of them in detail.

3.1 Goal 1: To develop a platform for continuous sensor data collection with annotations

Collecting data in natural environments along with its annotations is a challenging process that has been discussed in previous chapter for which it has been stated the need of smart annotation. However, there is a first need that must bee accomplished in order to achieve smart annotation and is to successfully collect data in natural environments. In order to collect data in natural environments a platform that allowed for its collection was needed. As it was mentioned before, one of the key problems on data annotation is the lack of adherence from users to annotate data. Therefore, such data collection platform had to be able to collect data for longitudinal studies while ensuring user's adherence.

This research goal aims at the development of a platform capable of running for long periods of time without supervision to enable longitudinal studies. The platform should continuously collect sensor data and provide an interface to the users that allows annotations to be made.

3.2 Goal 2: To evaluate the feasibility of smart data annotations for improved diet monitoring

In order to prove that smart annotations is achievable, this goal aims at proving that recognition models can be improved as more annotated data becomes available, but that such improvement can be maintained if the annotated data is intelligently selected. The goal is to show that even with reduced annotations to ensure user's adherence, it is still possible to achieve the same accuracy as if the annotations were indiscriminately requested. This goal focuses on diet monitoring through smart data annotations and model's retraining processes, particularly in the accurate detection of eating moments and additional activities that may correlate to eating patterns.

3.3 Goal 3: To run and evaluate the performance of deep learning models in wearable devices with reduced computing power

As the annotations are expected to occur in natural environments, the platform's main component is a wearable device from which data and annotations are taken. However, wearable devices are often constrained in computing power. Therefore, it is important to understand such constraints and to identify the limits of running recognition algorithms, in particular deep learning algorithms, wearable devices. This goal focuses on the evaluation on the impact that running deep learning models has in the wearable component of the data collection platform.

4. METHODS

In this chapter, the methods that were followed in this thesis are described. Each of the following sections describe the methods that correspond to each of the research goals that have been previously discussed in Chapter 2.

4.1 Data collection platform

In order to accomplish longitudinal data collection as described by our research goal 1 described in Chapter 3, one of the main concerns was to keep high user's compliance levels. This motivated our design decisions to be towards a wearable solution. Today's smartwatches are capable of collecting multiple sensor data such as IMU data, and heart rate through photoplethysmography (PPG) [35]. Additionally, most smartwatches are equipped with Bluetooth technologies which makes them suitable to connect with additional sensors such as chest straps which can provide additional sensor information such as breathing rate [36], increasing the range of possible sensor data that can be collected on the device. On the other hand, smartwatches are considered off-the-shelf devices meaning that they are greatly available for general public usage [37], making them a perfect tool to reach out people more easily. Therefore, given the sensing potential of smartwatches, as well as their pervasiveness in today's lifestyle, we decided to use a smartwatch as a main component of our data collection platform.

Despite all the positive features we have described about smartwatches, there is one important weakness that we must consider and is their reduced computing power. This restrictions plays an important role as in our work the retraining of deep learning models is something that constantly happens, and the training of deep learning models is well known for needing high computing power for training [38]. In order to accomplish a successful retraining process of recognition models, we needed to somehow perform these highly resource demanding tasks offline. Therefore, the platform has been designed in such a way that it can communicate with an external computer for offline processing.

The platform design is shown in Figure 4.1. For this particular work, the smartwatch device has been chosen to be a Polar watch model M600 as it's wearable component, and an Intel Next Unit of Computing (NUC) as their external server component. Despite the Polar watch being chosen because of its storage capacity, power efficiency, and features such as water resistance, it is important to mention that it can be replaced by any other Android based smartwatch as needed. The smartwatch can collect internal sensor data such accelerometer and gyroscope data provided by the IMU, as well as heart rate data provided by the PPG sensor. However, the smartwatch can also collect data from external sensors that provide data through Bluetooth technology using interfaces that allow a proper communication with the watch. Additionally, the smartwatch also captures BLE addresses from nearable devices with aims at collecting additional information that can be potentially used to aid in the annotation process. For example, BLE beacons can be placed at specific locations which can serve as a low energy alternative for location detection [39] instead of Global Positioning System (GPS) data as it tends to consume a lot of energy and sometimes users might feel that their privacy is compromised [40, 41, 42]

The smartwatch runs an Android app that is in charge of orchestrating the data collection processes, annotations, and recognition algorithms. The app has been designed to be versatile and extensible, allowing user's to collect and annotate data at will. An example of user's interaction with the app is shown in Figure 4.2, however, additional user interfaces were designed to suit different research needs that allowed GPS data collection and annotations through audio recordings. The app, intelligently manages the collection of sensor data to ensure power efficiency. There is also a notification module that is used to ask users for annotations based on rules that are set up on the watch such as timers or the decision of an intelligent annotation algorithm.

The proposed platform has been designed to be able to continuously collect data for an average of 10-12 hours. The idea of its usage is that user's use the smartwatch during the day to collect and annotate data. Then, at the end of the day the user plugs in the watch into the NUC for charging which activates the data transfer between devices. Then, the models are retrained overnight and transferred back to the smartwatch to have them updated for the next morning. Despite in this

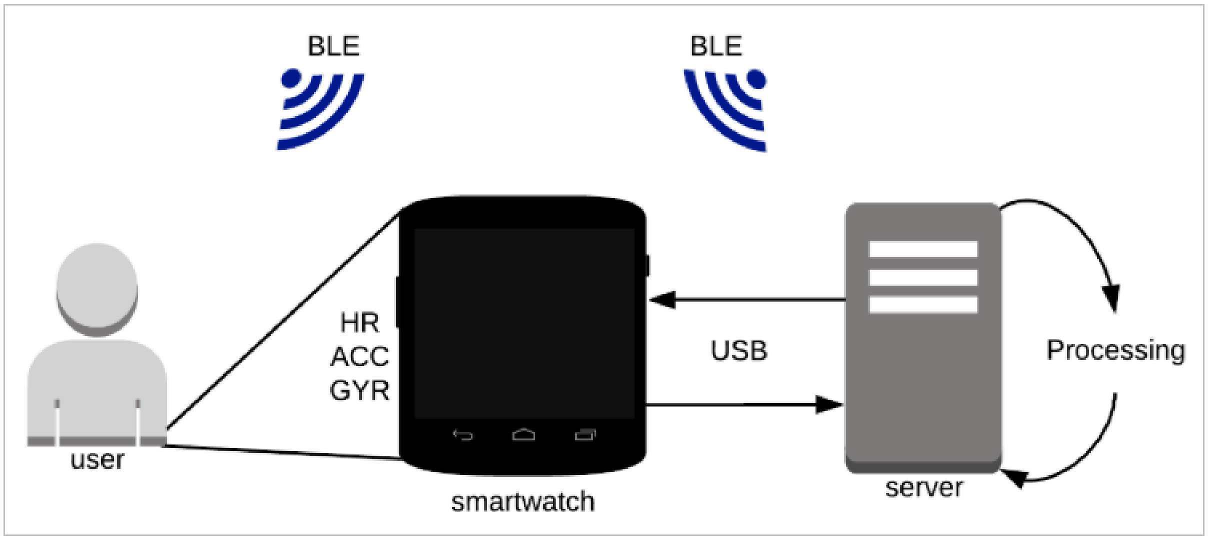

Figure 4.1: The proposed platform for data collection composed of a smartwatch for sensor data collection and an external server for high resource demanding computing processing. The smartwatch collects IMU and HR data from its internal sensors and also captures nearable BLE addresses. The server processes the information collected by the watch and updates the algorithms that are to be used by the watch.

particular work the platform is tested in a diet monitoring study, it has been designed to be scalable and easily adaptable to other research areas. For example, the platform can be used for remote cardiac rehabilitation monitoring for patients with cardiac conditions on the the smartwatch can track the user's activity and help to keep track of the user's performance throughout their rehabilitation program. The platform can potentially help to enhance continuous glucose monitoring (CGM) for patients with diabetes by providing activity data as context information to the glucose levels to better understand its changes.

4.2 Case study: Diet monitoring

In order to accomplish goals 2 and 3 described in Chapter 3, we wanted to perform the analysis on a real case scenario. Being diet monitoring an important matter as it helps for better nutrition and diseases management such as in diabetes management, we performed a diet monitoring study on which our goal was to improve eating moment detection through an intelligent data annotation process. In order to accomplish this, we wanted to not only be able to detect eating moments, but

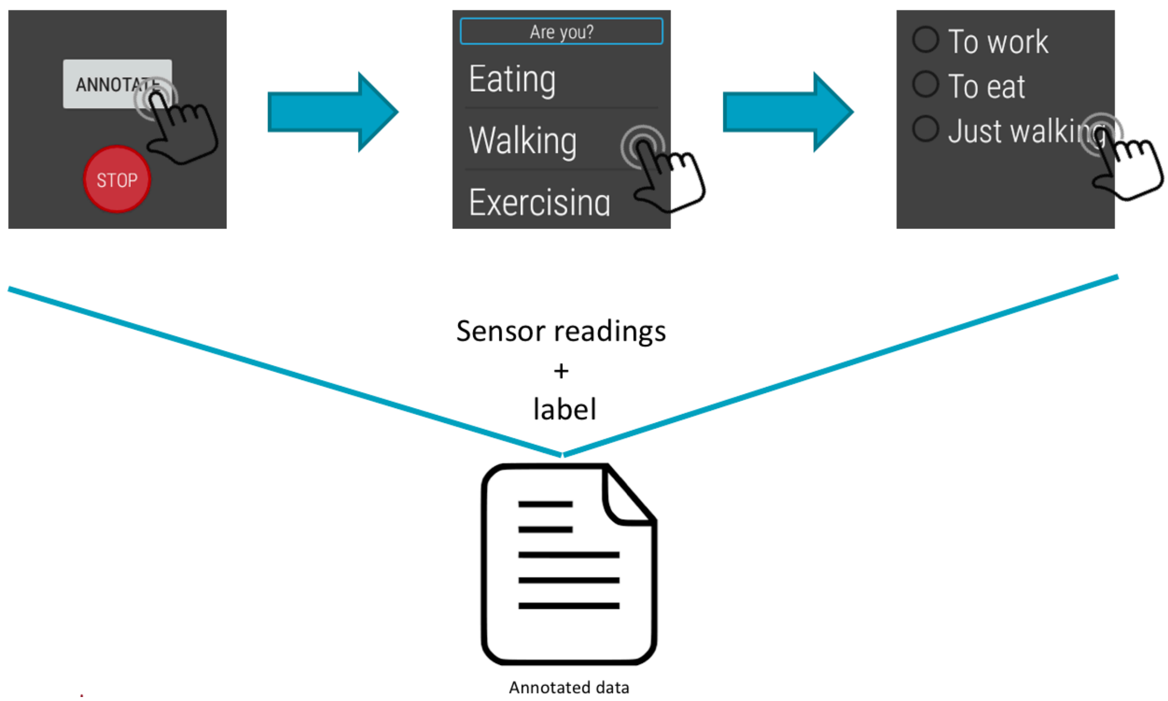

Figure 4.2: The Android based app that allows users to initiate the data collection process, and the lets them annotate data at will. From left to right we can see an annotation process as follows. First, the user decides to annotate data (can be because of a reminder based of a smart annotation algorithm). Then, the user selects the activity that is performing. Finally, the user can be prompt for additional information. The result is an annotation for the data being collected by the sensors.

also additional activities which might give a better insight of user's patterns of behavior in correlation to their eating patterns. In this study, 5 participants were given a smartwatch and they were asked to annotate whether they were eating or doing any other activity every 15 minutes for ground truth data collection and approximately 1 month of data was collected from each participant. The activities that were considered in this study are shown in Table 4.1. Through this study we aimed at using deep learning models for automatic feature extraction, and at understanding the constraints of running them on our smartwatch device.

This study was reviewed and approved by the Texas A&M University Institutional Review Board (IRB #2018-0998D). Therefore, informed consents were given to the participants and they were asked to sign them before participating in the study.

Activities				
Eating	Sitting	Walking	Exercising	Other

Table 4.1: The table shows the activities that were predicted in this work, where **Other** represents the unknown class.

4.3 Intelligent annotation

In order to accomplish goal 2 as discussed in Chapter 2, we first needed to demonstrate that there was a need for personalization. Then, we needed to demonstrate that deep learning algorithms can actually automatically extract features and recognize activities out of raw sensor data provided by the smartwatch. We then needed to demonstrate that we can in fact improve the model's performance through personalization accomplished by re-training models with more annotated data as it gets available. However, we also needed to demonstrate that such given improvements were still possible even through reduced annotations when such were intelligently obtained from users. The approaches we followed in order to accomplish each of the aforementioned needs are described in the following subsections.

4.3.1 Evaluating the need for personalization

In order to show that regardless of the type of activity being the same, different users tend to perform the same activity differently. It could be because each activities' restrictions on each user's lifestyles, or just because they performed differently by nature. Regardless of the reason, some models might not perform well in some subjects because of this particularities. We evaluated the raw sensor data collected from participants, being this the accelerometer and gyroscope data. We looked at multiple ways of data visualization that allowed us to understand how the data looks like for each subject.

In order to understand such subject's specific differences we looked at the mean value of each sensor channel for each subject. We also visualized the magnitudes of accelerometer and gyroscope data which gives a general perspective of the intensities captured by each given sensor. The magnitudes of accelerometer and gyroscope were calculated by Equation 4.1 and Equation 4.2

respectively. Where, the subscripts x, y, z indicates the channel for each given sensor reading.

$$acc_{mag} = \sqrt{acc_x^2 + acc_y^2 + acc_z^2} \tag{4.1}$$

$$gyro_{mag} = \sqrt{gyro_x^2 + gyro_y^2 + gyro_z^2} \tag{4.2}$$

4.3.2 Activity recognition through neural networks

As described in previously, we aimed at detecting eating moments and some activities that might correlate to it (See Table 4.1). Usually, feature engineering is performed on the raw sensor data and then, the extracted features are used to train the recognition models [43, 44, 45]. However, this approach requires some expert knowledge on the type of information being processed, and the manually extracted features may not necessarily serve as good predictors for a given activity [45]. In other words, important information contained in the raw sensor data might get ignored in the feature extraction process. Therefore, we wanted to get one step ahead and avoid this manual feature extraction process to avoid losing any information from the collected data.

In order to obtain the most out of the raw data, we opted for training neural network based models. In particular, we looked into convolutional neural networks (CNN) and long short-term memory (LSTM) networks which are two types of networks that can abstract information from raw data for automatic feature extraction [46]. The training process of neural networks is well known for being computationally demanding, however, running inferences is not necessarily as demanding as it can be as simple as a multiplication of weights. Therefore, since we already decided that any further training and processing of the data would be performed offline in a server as described in Figure 4.1, neural networks seemed like the perfect fit to our problem as inferences can be run in devices with limited processing power. Additionally, by using CNN and LSTM we can avoid any extra pre-processing of data that might be needed in order to manually compute the features that will be feed to any given model.

To this end, we had proposed 4 different model architectures for different automatic feature extraction with aims at finding the model that better fits our needs. Figure 4.3 describes in an

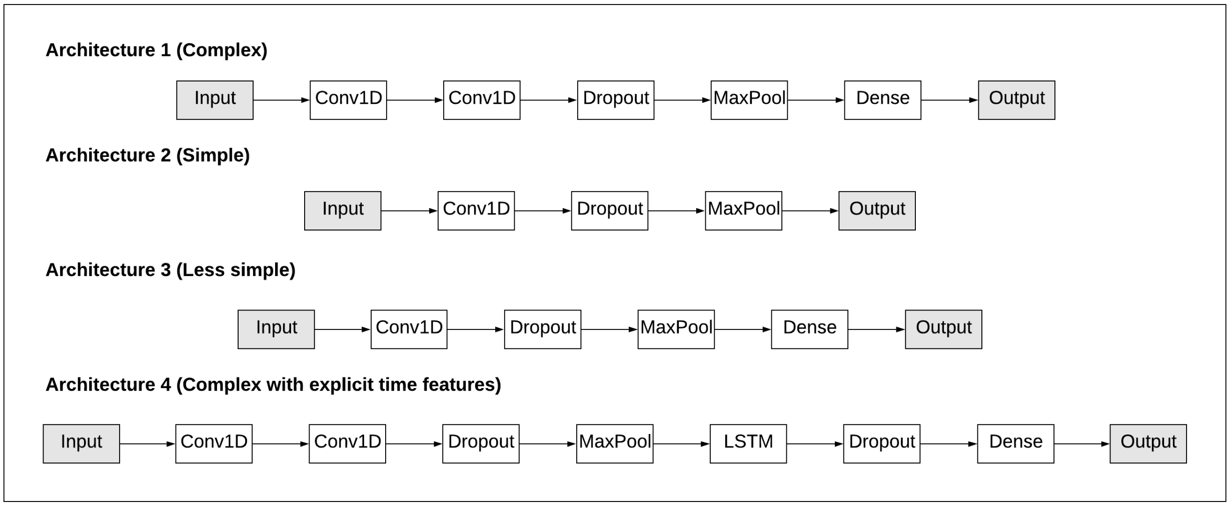

Figure 4.3: Architecture 1 designed to extract low and high level features. Architecture 2 and 3 designed as an alternative to architectures 2 and 3 with aims at reducing the model's parameters. Architecture 4 was designed with aims at explicitly extracting time related features.

intuitive way the architecture designs that we proposed. The architecture 1 is one of the most commonly used in CNN approaches. The first two convolutional layers allow us to extract low and high level features respectively, then the Drouput layer was added to reduce the model's likelihood of overfitting to the training data. The MaxPooling layer was added in order to reduce the models parameters complexity which is from fundamental importance as the final number of model's parameters correlates to the computation power that will be needed for running inferences. Architecture 2 and 3 had been proposed as an alternative to the complex architectures 1 and 3, the only difference between the two being the last dense layer in order to reduce the number of the model's parameters. In terms of the Dropout and MaxPool layers, the reasoning behind is the same as for architecture 1 and architecture 3. Architecture three is similar two architecture 1, with the difference that here we explicitly expect the network to extract time features through the LSTM layer.

After defining the architectures, we then needed to select the one that better fit our needs. Therefore, we trained each of the architectures using the data collected in the diet study. In order to feed the data to the models, we used a sliding window of 6 seconds with 50% overlap on the

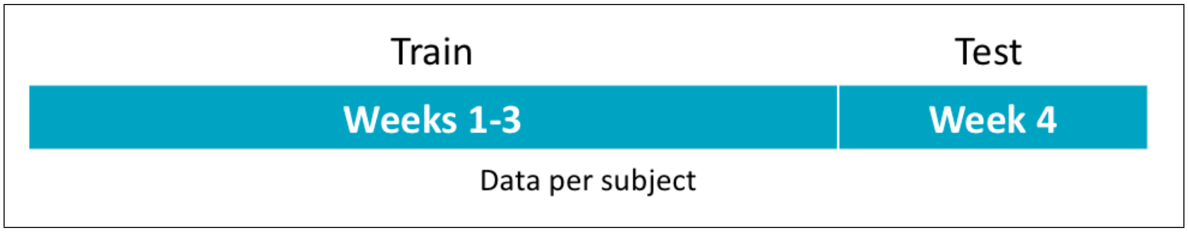

Figure 4.4: The first weeks of data were used as training set, and the last week as the test set to measure our model's accuracy.

6 sensor channels from accelerometer and gyroscope. Each window of data was then assigned its proper label and the window of data along with its label was feed into each model. Then, in order to select the best architecture to predict both our target activities and eating moments, we tested their performance on the last week of data when training on the first weeks. The train and test sets are depicted in Figure 4.4. For each architecture, we evaluated different batch sizes and epoch numbers and selected the architecture that showed best performance. After selecting the architecture that better suited our purposes, we performed hyperparameter tuning following a grid-search approach on which we evaluated different parameters for each of the layers.

4.3.3 Smart annotation and model personalization

In order to accomplish goal 2 as mentioned in Chapter 3, we wanted to show that personalization was possible through user annotated data and that by adding more annotations as we get them we can actually improve the model's performance. We also wanted to show that we could selectively decide whether a data sample needed to be annotated or not while maintaining accuracy, in order to reduce user's annotation burden.

To simulate the availability of new user's annotations in our offline analysis we divided each subject's data in 3 datasets as depicted in Figure 4.5. The first dataset was used to train a baseline model. The second dataset, was used as a simulation for annotation's availability. The third dataset was used to test the performance of the models as they are retrained with the availability of new data obtained from the second dataset. The process is described as follows, we first trained a model on the baseline dataset and tested against the test set. Then, in order to evaluate the impact of getting

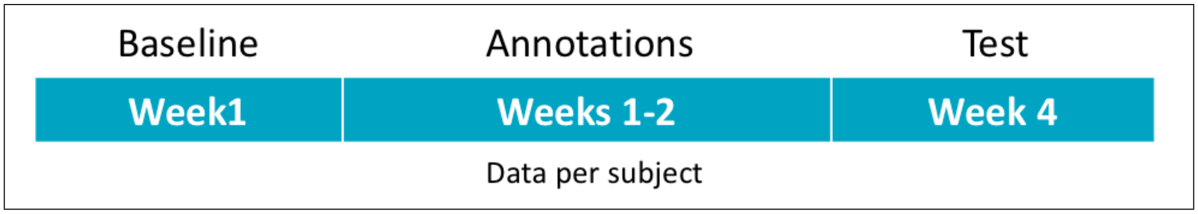

Figure 4.5: The first week of data was used to train a baseline model. The corresponding to the weeks between the first week and the last week was used to simulate annotations provided by the user. Finally, the last week of data was used to test the models.

multiple annotations, we progressively added annotated data from the annotations dataset in a daily basis and captured the model's accuracy. We wanted to show that the more data we get from the user, the better we get at predicting their activities.

On the other hand, we also wanted to minimize the need for annotations through an intelligent selection of data samples. On the retraining process of the baseline model, instead of adding all the data for retraining, we selected important data samples based on the model's output probability. First, we took the probability of all the samples in the train set per class that gets returned by the baseline model and take stored the average. Then, when a new sample was available we predicted its label with the current model and if the probability for that class was not within 1 standard deviation from the average, we simulated the need for it's annotation by simply adding the data sample to our train set to retrain the model.

4.4 Online evaluation

The third goal of this thesis mentioned in3 was to evaluate the implications of running deep learning models in devices with limited computing power such as are the smartwatches. It is well known that training neural networks is usually a resource demanding task. Running model's inferences is a whole different story, but still requires some computing power. We wanted to evaluate the CPU usage and energy impact of running our best model with different hyperparameter configurations. Since running inferences involves the usage of our model's parameters, the amount of operations needed to perform an inference greatly depends on the number of parameter, which depend on the hyperparameter configurations such as number of filters in a convolutional layer, or the number of neurons in a dense layer. Three different hyperparameter settings were evaluated considering, low, medium, and a high number of parameters. The CPU and energy impact were monitored and recorded through Android Studio's monitoring tool.

5. RESULTS

5.1 Data collection platform

The proposed data collection platform described in Section 4.1 has been tested in a real case scenario in this thesis through a diet monitoring study. The users were asked to wear the smartwatch for a whole day during approximately 1 month. The platform was able to successfully collect the data from users and let them annotate it at will using the designed user interface. In a previous work on which a custom made score for annotation requests was proposed, the platform was also used as the data collection platform. The platform was able to provide a tool for researchers to collect and annotated data through a custom made user interface that additionally allowed to record audio, and GPS data at will [47].

5.2 Intelligent annotation

In this section, we discuss the results obtained on the methods discussed in Section 4.3

5.2.1 Evaluating the need for personalization

We evaluated different ways on which we could visualize the need for personalization in an intuitive way. First, we looked at the mean and variances on the sensor channels per each activity. The goal was to be able to see differences on the average values on each channel per user on both the accelerometer data as seen in Figure 5.1, and gyroscope data as seen in Figure 5.2. However, it was not clear if there were actually differences between users, at least not in all sensor channels. This motivated another approach on which instead of looking at the mean and variances of each sensor channel, we looked at the same statistics on the magnitude on each sensor calculated through equation 4.1 and equation 4.2. This time, the differences between activities were a little clearer. However, this was still not an intuitive way of noticing user's need for personalization.

Another approach we followed was to 3d plot the accelerometer and gyroscope raw data per activity per user, however, given the amount of data samples corresponding to each activity, it was not easy to distinguish any clear differences between subjects. In order to give this approach a

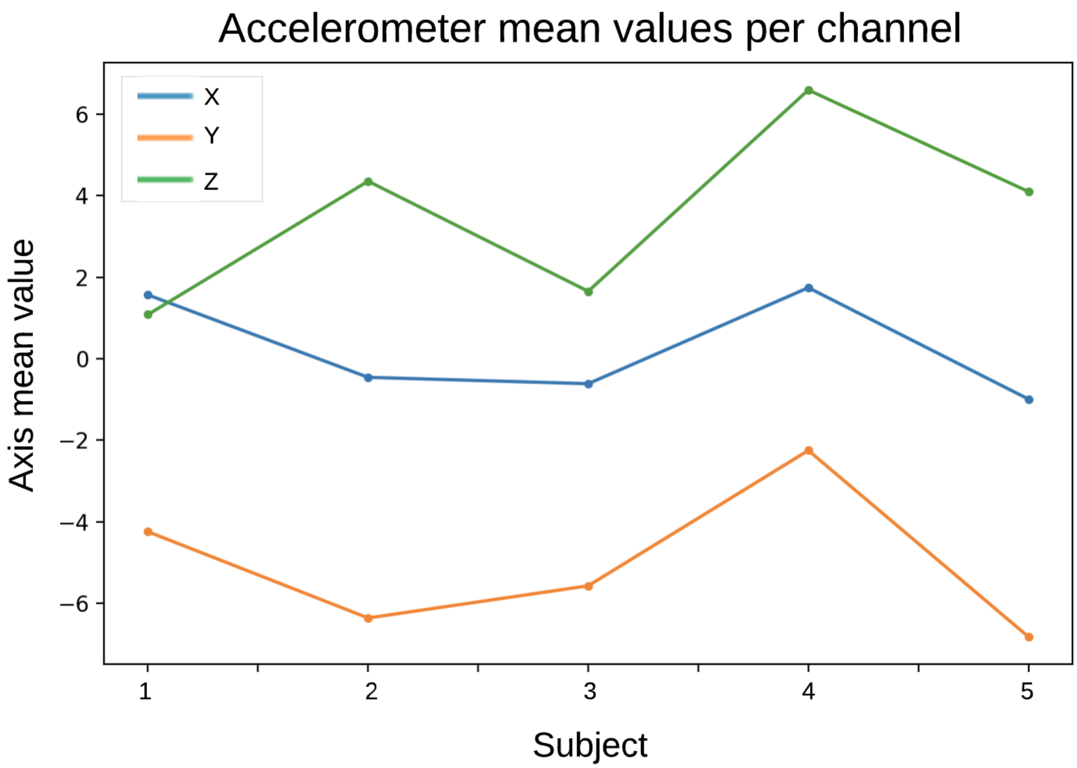

Figure 5.1: The figure shows the mean values per accelerometer channel per each subject.

spin, we decided to 2d plot the magnitude data from the accelerometer and gyroscope per subject. This approach resulted in a more intuitive visualization as can be seen in Figure 5.3 which shows the results obtained by plotting the accelerometer magnitude versus the gyroscope magnitude on three subjects for the sitting while working activity.

5.2.2 Activity recognition through neural networks

We followed the approach described in Section 4.3.2 for the architecture selection (See Figure 4.3). Figure 5.4 shows the results when training each architecture with different batch sizes. This process helped us to identify which from the 4 architectures was the one more likely to learn good features and predict our activities successfully. We observe that there are two architectures that perform better than the others, and are architectures 1 and 2. As we recall, these architectures were supposed to extract more complex features at the expenses of having more model's parame-

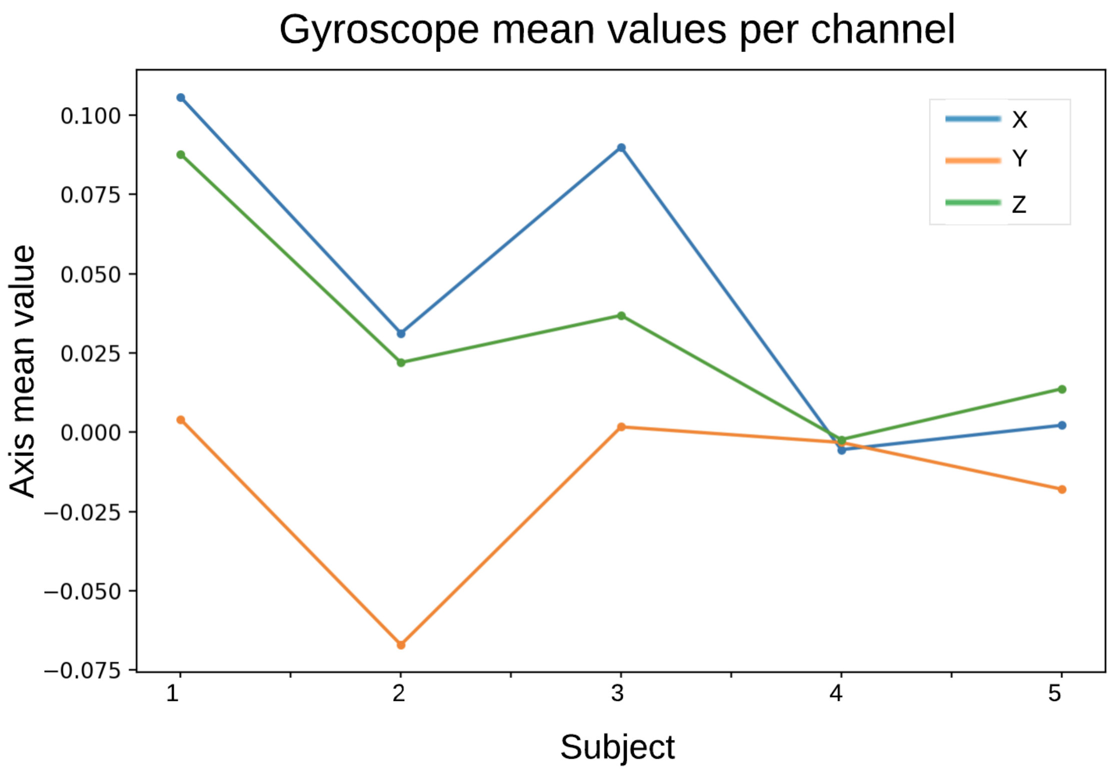

Figure 5.2: The figure shows the mean values per gyroscope channel per each subject.

ters, so it is not surprising that they perform better than the others. In an attempt to minimizing the number of final model's parameters, we proposed architectures 2 and 3. As for architecture 2, we note that the model was not even able to learn through the epochs. On the other hand, architecture 3 was able to learn but it was not able keep up improving and stopped learning at some point. Therefore, our final decision was to be made between architectures 1 and 2.

As we recall, our end goal was to run these types of architectures in a smartwatch. Therefore, besides accuracy, another important factor to consider on this selection process was the number of final parameters needed per architecture. As mentioned before, the final number of parameters plays an important role on the number of operations needed for inferences, which ultimately results in more processing power. Considering this last requirements, we decided to go for architecture 1, as it showed similar performance as architecture 4 both in terms of accuracy and f1 score. However, as architecture 4 has one more layer (the LSTM layer), it implies more parameters and therefore,

Figure 5.3: The figure shows the magnitude of accelerometer and gyroscope data corresponding to the activity sitting while working. Three subjects are shown in this figure. We can see that despite subject 1 and subject 2 seemed to share similar patterns, subject 3 seems to be clearly different and has its own patterns. This visualization suggests that users might perform activities differently.

more processing was going to be needed.

After having the architecture selected, we then performed a grid-search approach for hyperparameter tuning. Considering the most important hyperparameters found in the layers for architecture 1, we list the covered hyperparameters in Table 5.1 and tested all the possible combinations resulting in a total of 144 models being evaluated. Figure 5.5 shows the best 5 hyperparameter configurations for architecture one.

5.2.3 Smart annotation and model personalization

As we discussed in Section 4.3.3, we wanted to improve our recognition models through retraining processes motivated by the availability of new annotated data samples. Recalling that we had 3 different we progressively added, trained, and tested models by adding data from the

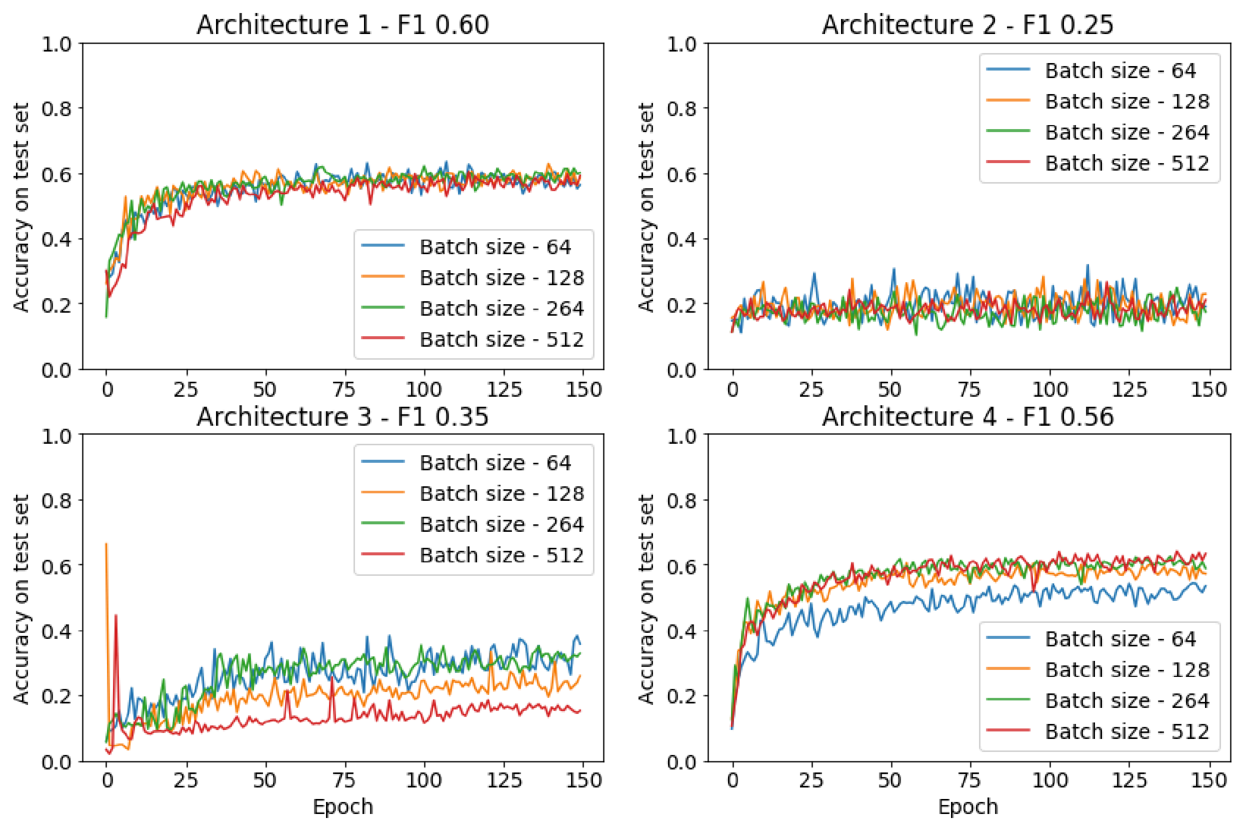

Figure 5.4: The figure shows the accuracy results and f1 score (micro-averaged) for each of the different architectures on different batch sizes. We note that architectures 1 and 4 perform better than architecture 2 and 3 as they are more complex and therefore extract better features out of the raw sensor data.

annotations dataset in a simulated daily basis. In other words, after adding the data collected during one day, we would retrain the model and test it and repeat this with the following day. In a first instance, we wanted to evaluate the models accuracy levels by assuming we would have unlimited annotated data by the user, which was the same as simply adding all the available data within a day under the annotations dataset. Figure 5.6 shows the AUROC values for each of the activities that we were trying to classify. As we observe, it is hard to tell if there is any consistency on the AUROC values as we add more data. We observe that for eating moments, there is a clear improvement when we add data from days 1 through 6, however a significant drop appears after the data for day 7 is added. Similarly, for sitting there is a decrease when the data from the last day gets added. For walking, rather than showing any improvement when adding more data, it looks

Hyperparameter	Values
Number of filters ConvLayer 1	128, 64, 32
Number of filters ConvLayer 2	128, 64, 32
Kernel size ConvLayer 1	3, 6
Kernel size ConvLayer 2	3, 6
Dropout rate DropoutLayer	0.4, 0.6
Number of neurons DenseLayer	80, 120

Table 5.1: The table shows the hyperparameter values that were considered to evaluate for architecture 1 hyperparameter tuning.

like it is getting worst.

So, why would it be that some activities are improved, but some others are not? It could be multiple things, for example, it could be possible that the 6 seconds window we have chosen is using up portions of data with both motions related to the activity of interest, and other motions that might not be part of the activity itself. For example, when eating out of 6 seconds maybe only 3 represent the motion of taking a spoon to the mouth, and the other 3 the actions of taking the spoon. It could also be that the users have performed the same activity in a different way over time, which is not necessarily wrong, it is simply that we might need to collect more data for more time to be able to cover such cases. We also believe that the reason behind the performances' drops might be because not all the samples add value to our models, which suggests that we should also consider which samples should we use rather than simply and blindly adding such results. This aligns with our proposed approach for sample selection.

As described in Section 4.3.3, instead using every single sample collected through a day in the annotations dataset, we only used the ones that we believed could contribute the most to the training set. So, after training the baseline model we computed the average probability output of our baseline model for each given class on the training set. When a new sample arrived and its probability was estimated by the model, we checked if the probability was within 1 standard deviation from its corresponding mean previously calculated. If the data was within 1 standard deviation, we assumed that the new sample was not very different from what we have seen so far,

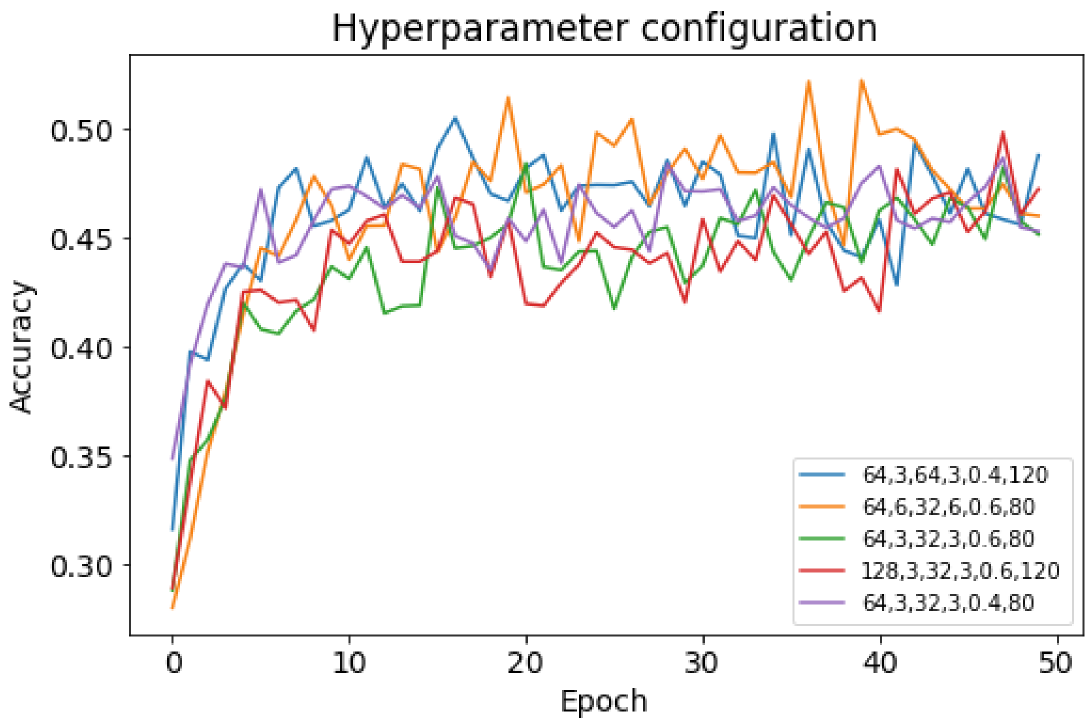

Figure 5.5: The figure shows the 5 configurations with highest accuracy. In the legend are the values for each of the parameters and follows the next order: Number of filters ConvLayer 1, Number of filters ConvLayer 2, Kernel size ConvLayer 1, Kernel size ConvLayer 2, Dropout rate DropoutLayer, Number of neurons DenseLayer.

therefore, it might not be worth annotating and using for retraining our model. On the other hand, if the probability output was outside 1 standard deviation from the mean, we assumed that this new sample was considerably different from what we have seen so far, therefore, it may be worth annotating and using for retraining. The results are shown in Figure 5.7. Despite the solution stabilizes some activities, it doesn't work for all of them suggesting that the intelligent selection is possible but it still needs further analysis. For example, it could be that we might need to create multiple models per user's activity and sub-activity based on the details they provided. In other words, rather than having a general model for sitting, we might need to have a model for sitting-while-working, and one for sitting-while-watching-tv.

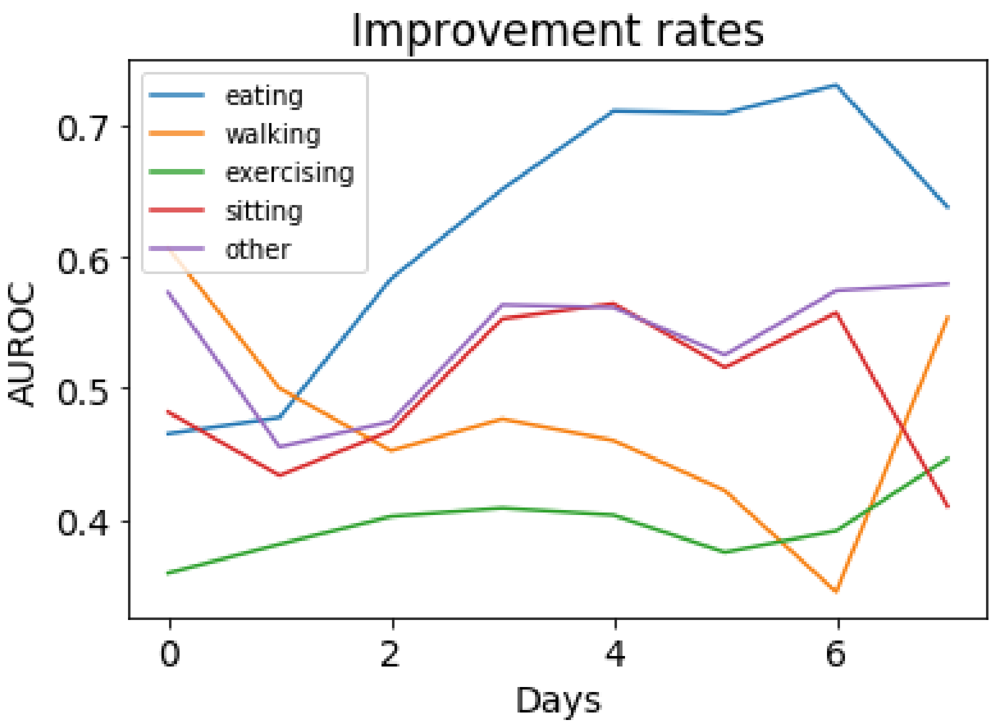

Figure 5.6: The figure shows the AUROC values per activity for subject 3. The values corresponding to day 0 are the ones obtained from evaluating the baseline model as it is against the test model. Then, x represents the number of day that was added up to that point (considering the past have already been added).

5.3 Online evaluation

After all the offline processing done so far, we needed to analyze what would it imply to run our deep learning models in a device with such reduced computing power such as the smartwatch. Therefore, we designed an experiment on which we evaluated the CPU usage and energy usage on the smartwatch in 3 different architecture configurations as described in Section 4.4. We have integrated a new module in our Android app running in our smartwatch for running model's inferences every 5 seconds.

In order to evaluate the impact of running inferences, we first needed to have a baseline to compare against to. Therefore, we first evaluated the usual CPU and energy usage of our app without running any inference. We used Android Studio's built in tool for monitoring called **Android Profiler** that gives estimates on the CPU usage and the impact that our app has in terms of energy

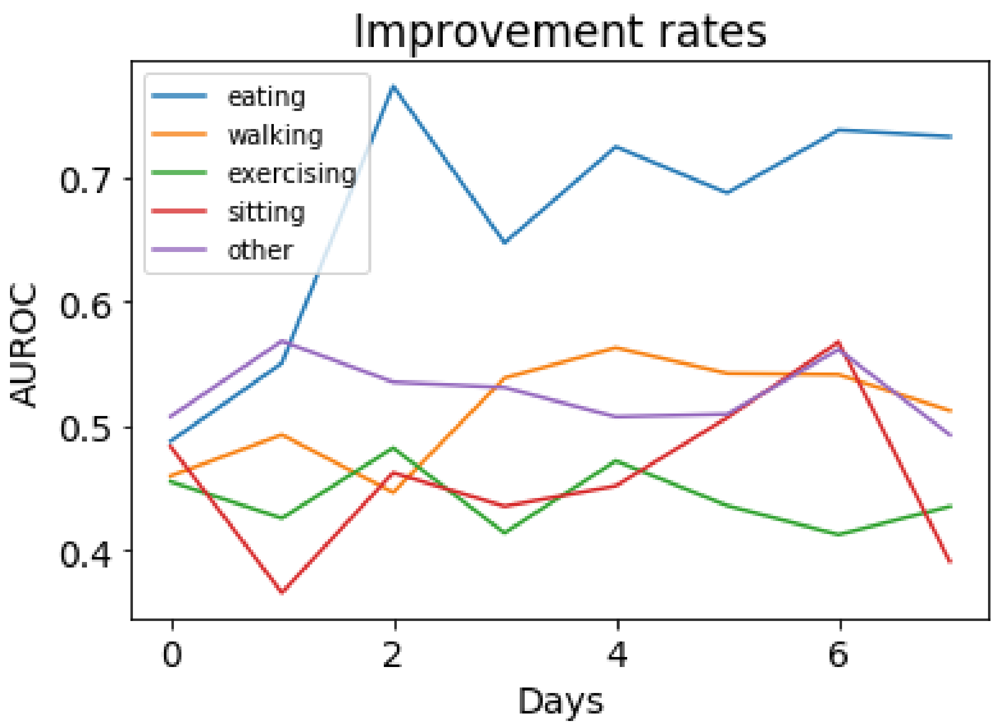

Figure 5.7: The figure shows the AUROC values per activity for subject 3. The values corresponding to day 0 are the ones obtained from evaluating the baseline model as it is against the test model. Then, x axis represents the number of day that was added up to that point (considering the past have already been added).

consumption. Figure 5.8 shows the impact that our app regularly has. We note that on average, the CPU usage is around 5% and the energy impact has been categorized by Android Profiler as **light**.

Then, we wanted to evaluate how much would it impact running inferences in the smartwatch, but more importantly, how this impact changes based on the model's hyperparameters configuration. Figure 5.9 shows the CPU and energy impact of running inferences in the smartwatch with different configurations. The light configuration is the architecture 1 being trained with a small number of filters and neurons. The medium configuration is the architecture 1 being trained with some more filters and neurons. As for the heavy configuration, it is the architecture 1 being trained with even more filters and neurons. The goal was to show that, the hyperparameter configuration correlates to the final number of parameters each model will end up having after training, which also correlates to the CPU and energy impact. Despite in Figure 5.10 we can see that Android

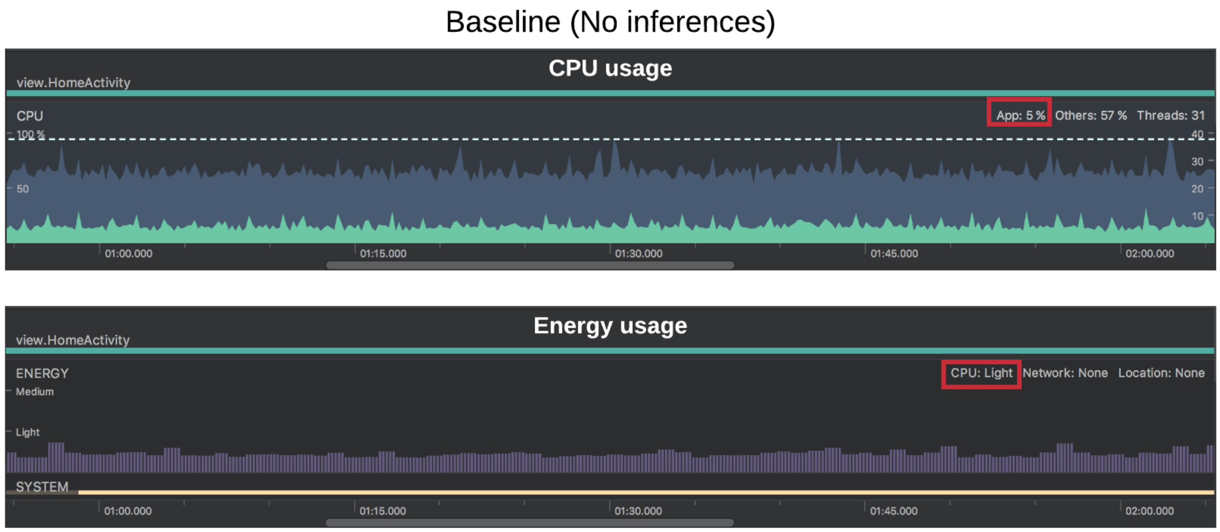

Figure 5.8: The figure shows CPU and energy consumption when running our data collection app in the smartwatch with no inferences, simply collecting data. The average CPU usage was 5% and the average energy impact was categorized as **light**.

Profiler has categorized the energy consumption as **light**, there is a slight increment on the energy levels depicted by the purple barplots. This will add up with time and will clearly reduce the up-time of the app for collecting data. This is clearer when we look at Figure 5.9 on which we observe that the more parameters we have, the heavier the computation will be. Therefore, this suggests that the hyperparameter configuration is something that we should also consider when training our models and should be as important as paying attention to our model's performance to ensure the platform runs for extended periods of time.

After analyzing the information obtained by running the online experiments, we wanted to get a better insight on what to expect in terms of our data collection. In other words, how would this impact the overall data collection process? The smartwatch is able to continuously collect data for approximately 10-12 hours. Knowing this information, and with our new findings we have created a small projection to get the new estimated maximum duration our smartwatch would be able to continuously collect data. This estimation is shown in Table 5.2

Despite our estimations might not seem encouraging, there are still multiple ways on which we can improve the app's efficiency and maximize the total time we can continuously collect data

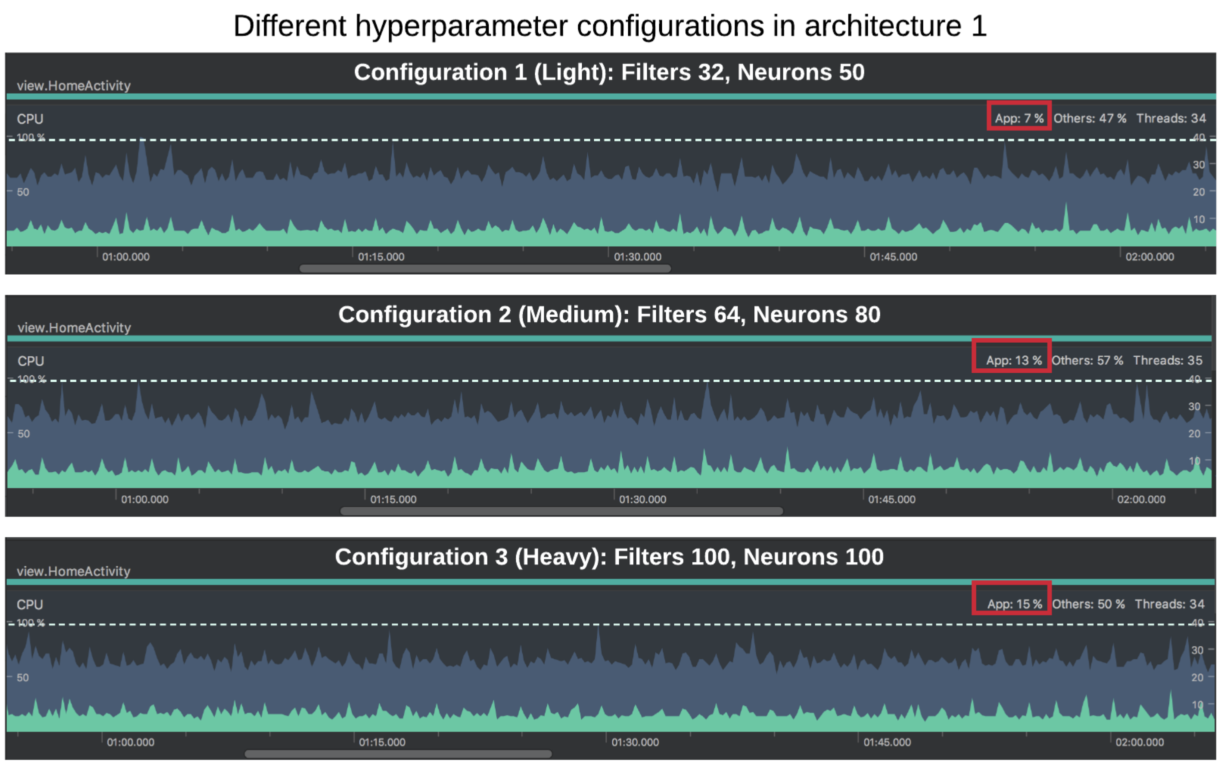

Figure 5.9: The figure shows CPU consumption when running our data collection app in the smartwatch while running inferences with different hyperparameter configuration settings identified as low, medium, and heavy referring to the final number of parameters each trained model will have. We notice a clear trend on the CPU usage as it increases as the number of parameters increases, which correlates to the hyperparameter configuration.

even while we run inferences on the smartwatch. We are currently running inferences constantly, but we could instead run them only when needed. For example, inferences can be run every time the accelerometer data has a significant change. Given that some activities are stationary, the sensor readings for both the accelerometer and gyroscope might remain somewhat constant. Why would we need to run an estimation if the sensor readings have not changed in a while? it is very likely that we are still performing the same activity. Another approach could be to only run inferences when specific activity levels are met considering the magnitude of the accelerometer and gyroscope. If there is no significant activity level, why running inferences? Those are just some possible approaches in terms of inferences, however, there are also a few other ways to improve the energy usage.

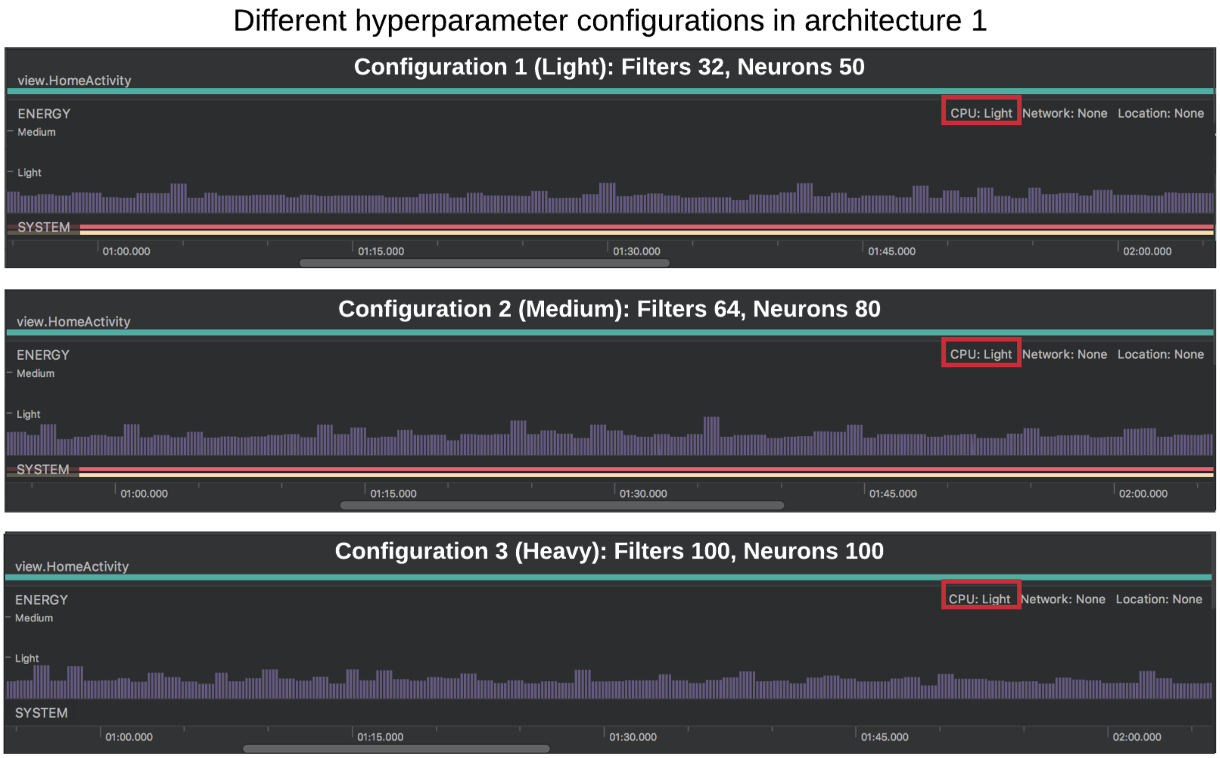

Figure 5.10: The figure shows energy consumption when running our data collection app in the smartwatch while running inferences with different hyperparameter configuration settings identified as low, medium, and heavy referring to the final number of parameters each trained model will have.

We also need to consider that the analysis we made had the smartwatch also capturing BLE addresses. This can also consume a considerable amount of energy specially when a huge number of devices is nearby. However, collecting this data continuously might not be necessary and we could opportunistically turn this BLE collection *on*/*off*. BLE addresses are being collected with the purpose of providing contextual information, however, such contextual information might not always be needed. Therefore, we could just turn this BLE data collection *on* when the model's output probability for activity recognition is low (as context can help to improve this), or when the probability for two or more classes is close to each other, the context provided by the BLEs can serve as a tie breaker.

Configuration	Average CPU %	Estimated duration
No model	5	10 - 12 hours
1	7	7.14 - 8.5 hours
2	13	3.8 - 4.6 hours
3	15	3.3 - 4 hours

Table 5.2: The table shows the new maximum duration estimation for data collection based on the CPU usage according to each architecture configuration.

6. CONCLUSIONS AND FUTURE WORK

In this thesis our main goal was to provide a data collection platform that enabled longitudinal studies which also possessed capabilities for smart data annotation. More specifically, we have defined 3 research goals described as follows:

1. Development of a data collection platform for longitudinal studies.

2. Analysis of a deep learning approach for personalization and smart annotation.

3. Performance analysis of a a deep learning approach on a wearable devices.

We have introduced a wearable based data collection platform with smart annotation capabilities for longitudinal studies that is also able to run deep learning models in real time. The platform is composed of a smartwatch for sensor data collection, and a server for offline data processing. The feasibility of the platform for data collection had been previously tested on a prior contribution [47] and was tested again in a second study on diet monitoring. However, the platform design is not exclusive of diet monitoring and can be applied in different research areas such as in cardiac rehabilitation, and enhanced glucose monitoring.

A diet monitoring study was performed on which 5 participants were recruited and were asked to collect annotated data through our smartwatches for approximately 1 month. The data collected during the diet monitoring study was used for further experiments on the detection of eating moments and model's personalization as well as for the analysis of a smart annotation technique. We analyzed different neural network architectures for automatic feature extraction and accurate detection of activities. We also presented an evaluation of the performance improvements obtained by intelligently requesting annotations for accurate detection of eating moments and activities such as sitting, walking, and exercising.

Finally, we have evaluated the feasibility of running neural network models' inferences in devices with reduced computing power such as are the smartwatches. In particular, we evaluated those neural network architectures that automatically extract features from raw sensor data. We found out that the impact on CPU and energy usage is dependent on the hyperparameter configuration of the models and that it is an aspect of architecture design that must be considered. However, the impact of running such models in an online fashion is still manageable and we discussed possible approaches to improving the data collection process' overall performance.

Further research is needed in terms of personalization through reduced user's annotations and on better and more sophisticated techniques for sample selection for deciding whether or not an annotation must be requested from the users.

Our platform enables for future work on behavior monitoring in different research areas such as in cardiac rehabilitation and diabetes management. User's can be monitoring through our platform to determine whether they are doing a satisfactory set of exercises based on their rehabilitation program, or their activities can be monitored in order to provide better context to their glucose levels for a more informed diabetes management. Our contributions in this thesis also opens the path for further research in the smart annotation process of data as we shown that it is feasible in a diet monitoring study.

Further research can be accomplished to improve smart automatic data annotation by investigating reinforcement learning approaches in order to learn user's patterns of behavior and personal information for improved adherence such as:

- Question asking rates
- Appropriate moments to request annotations
- Types of questions that better provide information to enhance the recognition models (detailed question asking)

REFERENCES

[1] E. Jovanov, "Preliminary analysis of the use of smartwatches for longitudinal health monitoring," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 865–868, IEEE, 2015.

[2] V. Ahanathapillai, J. D. Amor, Z. Goodwin, and C. J. James, "Preliminary study on activity monitoring using an android smart-watch," *Healthcare technology letters*, vol. 2, no. 1, pp. 34–39, 2015.

[3] E. Årsand, M. Muzny, M. Bradway, J. Muzik, and G. Hartvigsen, "Performance of the first combined smartwatch and smartphone diabetes diary application study," *Journal of diabetes science and technology*, vol. 9, no. 3, pp. 556–563, 2015.

[4] K. Van Laerhoven, M. Borazio, and J. H. Burdinski, "Wear is your mobile? investigating phone carrying and use habits with a wearable device," *Frontiers in ICT*, vol. 2, p. 10, 2015.

[5] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *International Conference on Pervasive Computing*, pp. 1–17, Springer, 2004.

[6] F. Foerster, M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," *Computers in Human Behavior*, vol. 15, no. 5, pp. 571–583, 1999.

[7] L. E. Burke, J. Wang, and M. A. Sevick, "Self-monitoring in weight loss: a systematic review of the literature," *Journal of the American Dietetic Association*, vol. 111, no. 1, pp. 92–102, 2011.

[8] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell, "Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*, pp. 3–14, ACM, 2014.

[9] H. M. Dalal, A. Zawada, K. Jolly, T. Moxham, and R. S. Taylor, "Home based versus centre based cardiac rehabilitation: Cochrane systematic review and meta-analysis," *Bmj*, vol. 340, p. b5631, 2010.

[10] J. C. Bollen, S. G. Dean, R. J. Siegert, T. E. Howe, and V. A. Goodwin, "A systematic review of measures of self-reported adherence to unsupervised home-based rehabilitation exercise programmes, and their psychometric properties," *BMJ open*, vol. 4, no. 6, p. e005044, 2014.

[11] M. Varnfield, M. Karunanithi, C.-K. Lee, E. Honeyman, D. Arnold, H. Ding, C. Smith, and D. L. Walters, "Smartphone-based home care model improved use of cardiac rehabilitation in postmyocardial infarction patients: results from a randomised controlled trial," *Heart*, pp. heartjnl–2014, 2014.

[12] N. D. Peterson, K. R. Middleton, L. M. Nackers, K. E. Medina, V. A. Milsom, and M. G. Perri, "Dietary self-monitoring and long-term success with weight management," *Obesity*, vol. 22, no. 9, pp. 1962–1967, 2014.

[13] L.-J. Li and L. Fei-Fei, "Optimol: automatic online picture collection via incremental model learning," *International journal of computer vision*, vol. 88, no. 2, pp. 147–168, 2010.

[14] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, ACM, 1998.

[15] A. Kapoor and E. Horvitz, "Experience sampling for building predictive user models: a comparative study," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 657–666, ACM, 2008.

[16] B. Longstaff, S. Reddy, and D. Estrin, "Improving activity classification for health applications on mobile devices using active and semi-supervised learning," in *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, pp. 1–7, IEEE, 2010.

[17] E. Thomaz, I. Essa, and G. D. Abowd, “A practical approach for recognizing eating moments with wrist-mounted inertial sensing,” in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 1029–1040, ACM, 2015.

[18] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber, “Smartwatch-based activity recognition: A machine learning approach,” in *Biomedical and Health Informatics (BHI), 2016 IEEE-EMBS International Conference on*, pp. 426–429, IEEE, 2016.

[19] B. Mortazavi, E. Nemati, K. VanderWall, H. G. Flores-Rodriguez, J. Y. J. Cai, J. Lucier, A. Naeim, and M. Sarrafzadeh, “Can smartwatches replace smartphones for posture tracking?,” *Sensors*, vol. 15, no. 10, pp. 26783–26800, 2015.

[20] A. Gaggioli, G. Pioggia, G. Tartarisco, G. Baldus, D. Corda, P. Cipresso, and G. Riva, “A mobile data collection platform for mental health research,” *Personal and Ubiquitous Computing*, vol. 17, no. 2, pp. 241–251, 2013.

[21] Y.-H. Kim, J. H. Jeon, B. Lee, E. K. Choe, and J. Seo, “Omnitrack: A flexible self-tracking approach leveraging semi-automated tracking,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. 67, 2017.

[22] A. Y. Barrera-Animas, L. A. Trejo, M. A. Medina-Pérez, R. Monroy, J. B. Camiña, and F. Godínez, “Online personal risk detection based on behavioural and physiological patterns,” *Information Sciences*, vol. 384, pp. 281–297, 2017.

[23] A. Filippoupolitis, W. Oliff, B. Takand, and G. Loukas, “Location-enhanced activity recognition in indoor environments using off the shelf smart watch technology and ble beacons,” *Sensors*, vol. 17, no. 6, p. 1230, 2017.

[24] T. Choudhury, G. Borriello, J. A. Landay, L. LeGrand, J. Lester, A. Rahimi, A. Rea, D. Wyatt, S. Consolvo, D. Haehnel, *et al.*, “The mobile sensing platform: An embedded activity recognition system,” *IEEE Pervasive Computing*, no. 2, pp. 32–41, 2008.

[25] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *International conference on pervasive computing*, pp. 158–175, Springer, 2004.

[26] D. Guan, W. Yuan, Y.-K. Lee, A. Gavrilov, and S. Lee, "Activity recognition based on semi-supervised learning," in *Embedded and Real-Time Computing Systems and Applications, 2007. RTCSA 2007. 13th IEEE International Conference on*, pp. 469–475, IEEE, 2007.

[27] Y. Zhou and S. Goldman, "Democratic co-learning," in *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pp. 594–602, IEEE, 2004.

[28] M. Fuller-Tyszkiewicz, H. Skouteris, B. Richardson, J. Blore, M. Holmes, and J. Mills, "Does the burden of the experience sampling method undermine data quality in state body image research?," *Body Image*, vol. 10, no. 4, pp. 607–613, 2013.

[29] A. Mannini and S. Intille, "Classifier personalization for activity recognition using wrist accelerometers," *IEEE journal of biomedical and health informatics*, 2018.

[30] L. Chan, V. D. Swain, C. Kelley, K. de Barbaro, G. D. Abowd, and L. Wilcox, "Students' experiences with ecological momentary assessment tools to report on emotional well-being," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, p. 3, 2018.

[31] C. Bi, G. Xing, T. Hao, J. Huh, W. Peng, and M. Ma, "Familylog: A mobile system for monitoring family mealtime activities," in *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*, pp. 21–30, IEEE, 2017.

[32] A. Bedri, R. Li, M. Haynes, R. P. Kosaraju, I. Grover, T. Prioleau, M. Y. Beh, M. Goel, T. Starner, and G. Abowd, "Earbit: using wearable sensors to detect eating episodes in unconstrained environments," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. 37, 2017.

[33] H. Kalantarian, N. Alshurafa, T. Le, and M. Sarrafzadeh, "Monitoring eating habits using a piezoelectric sensor-based necklace," *Computers in biology and medicine*, vol. 58, pp. 46–55, 2015.

[34] M. Farooq and E. Sazonov, "Real time monitoring and recognition of eating and physical activity with a wearable device connected to the eyeglass," in *Sensing Technology (ICST), 2017 Eleventh International Conference on*, pp. 1–6, IEEE, 2017.

[35] D. Phan, L. Y. Siong, P. N. Pathirana, and A. Seneviratne, "Smartwatch: Performance evaluation for long-term heart rate monitoring," in *2015 International Symposium on Bioelectronics and Bioinformatics (ISBB)*, pp. 144–147, IEEE, 2015.

[36] J. A. Johnstone, P. A. Ford, G. Hughes, T. Watson, and A. T. Garrett, "Bioharness multivariable monitoring device: part. ii: reliability," *Journal of sports science & medicine*, vol. 11, no. 3, p. 409, 2012.

[37] "Commercial off-the-shelf definition." `https://www.acquisition.gov/content/2101-definitions`. Accessed: 2019-05-03.

[38] A. Guzhva, S. Dolenko, and I. Persiantsev, "Multifold acceleration of neural network computations using gpu," in *International Conference on Artificial Neural Networks*, pp. 373–380, Springer, 2009.

[39] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[40] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pp. 105–114, ACM, 2010.

[41] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt, "Micro-blog: sharing and querying content through mobile phones and social participation," in *Proceedings of the 6th*

international conference on Mobile systems, applications, and services, pp. 174–186, ACM, 2008.

[42] P. Klasnja, S. Consolvo, T. Choudhury, R. Beckwith, and J. Hightower, "Exploring privacy concerns about personal sensing," in *International Conference on Pervasive Computing*, pp. 176–183, Springer, 2009.

[43] J. Shukla, M. Barreda, J. Oliver, G. Nandi, D. Puig, *et al.*, "Feature extraction and selection for emotion recognition from electrodermal activity," *IEEE Transactions on Affective Computing*, 2019.

[44] P. Dollár, Z. Tu, H. Tao, and S. Belongie, "Feature mining for image classification," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[45] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Aaai*, vol. 5, pp. 1541–1546, 2005.

[46] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Two stream lstm: A deep fusion framework for human action recognition," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 177–186, IEEE, 2017.

[47] R. Solis, A. Pakbin, A. Akbari, B. J. Mortazavi, and R. Jafari, "A human-centered wearable sensing platform with intelligent automated data annotation capabilities," in *Proceedings of the International Conference on Internet of Things Design and Implementation*, pp. 255–260, ACM, 2019.