

Clemson University

**TigerPrints**

---

All Dissertations

Dissertations

---

August 2020

## Detecting Periods of Eating in Everyday Life by Tracking Wrist Motion – What is a Meal?

Surya Sharma

*Clemson University, s@suryasharma.com*

Follow this and additional works at: [https://tigerprints.clemson.edu/all\\_dissertations](https://tigerprints.clemson.edu/all_dissertations)

---

### Recommended Citation

Sharma, Surya, "Detecting Periods of Eating in Everyday Life by Tracking Wrist Motion – What is a Meal?" (2020). *All Dissertations*. 2675.

[https://tigerprints.clemson.edu/all\\_dissertations/2675](https://tigerprints.clemson.edu/all_dissertations/2675)

This Dissertation is brought to you for free and open access by the Dissertations at TigerPrints. It has been accepted for inclusion in All Dissertations by an authorized administrator of TigerPrints. For more information, please contact [kokeefe@clemson.edu](mailto:kokeefe@clemson.edu).

# DETECTING PERIODS OF EATING IN EVERYDAY LIFE BY TRACKING WRIST MOTION - WHAT IS A MEAL?

---

A Dissertation  
Presented to  
the Graduate School of  
Clemson University

---

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy  
Computer Engineering

---

by  
Surya Prakash Sharma  
August 2020

---

Accepted by:  
Dr. Adam Hoover, Committee Chair  
Dr. Brian Dean  
Dr. Yingjie Lao  
Dr. Yongqiang Wang

# Abstract

Eating is one of the most basic activities observed in sentient animals, a behavior so natural that humans often eat without giving the activity a second thought. Unfortunately, this often leads to consuming more calories than expended, which can cause weight gain - a leading cause of diseases and death. This proposal describes research in methods to automatically detect periods of eating by tracking wrist motion so that calorie consumption can be tracked. We first briefly discuss how obesity is caused due to an imbalance in calorie intake and expenditure. Calorie consumption and expenditure can be tracked manually using tools like paper diaries, however it is well known that human bias can affect the accuracy of such tracking. Researchers in the upcoming field of automated dietary monitoring (ADM) are attempting to track diet using electronic methods in an effort to mitigate this bias.

We attempt to replicate a previous algorithm that detects eating by tracking wrist motion electronically. The previous algorithm was evaluated on data collected from 43 subjects using an iPhone as the sensor. Periods of time are segmented first, and then classified using a naive Bayesian classifier. For replication, we describe the collection of the Clemson all-day data set (CAD), a free-living eating activity dataset containing 4,680 hours of wrist motion collected from 351 participants - the largest of its kind known to us. We learn that while different sensors are available to log wrist acceleration data, no unified convention exists, and this data must thus be transformed between conventions. We learn that the performance of the eating detection algorithm is affected due to changes in the sensors used to track wrist motion, increased variability in behavior due to a larger participant pool, and the ratio of eating to non-eating in the dataset.

We learn that commercially available acceleration sensors contain noise in their reported readings which affects wrist tracking specifically due to the low magnitude of wrist acceleration. Commercial accelerometers can have noise up to 0.06g which is acceptable in applications like au-

tomobile crash testing or pedestrian indoor navigation, but not in ones using wrist motion. We quantify linear acceleration noise in our free-living dataset. We explain sources of noise, a method to mitigate it, and also evaluate the effect of this noise on the eating detection algorithm.

By visualizing periods of eating in the collected dataset we learn that that people often conduct secondary activities while eating, such as walking, watching television, working, and doing household chores. These secondary activities cause wrist motions that obfuscate wrist motions associated with eating, which increases the difficulty of detecting periods of eating (meals). Subjects reported conducting secondary activities in 72% of meals. Analysis of wrist motion data revealed that the wrist was resting 12.8% of the time during self-reported meals, compared to only 6.8% of the time in a cafeteria dataset. Walking motion was found during 5.5% of the time during meals in free-living, compared to 0% in the cafeteria. Augmenting an eating detection classifier to include walking and resting detection improved the average per person accuracy from 74% to 77% on our free-living dataset ( $t[353]=7.86$ ,  $p<0.001$ ). This suggests that future data collections for eating activity detection should also collect detailed ground truth on secondary activities being conducted during eating.

Finally, learning from this data collection, we describe a convolutional neural network (CNN) to detect periods of eating by tracking wrist motion during everyday life. Eating uses hand-to-mouth gestures for ingestion, each of which lasts appx 1-5 sec. The novelty of our new approach is that we analyze a much longer window (0.5-15 min) that can contain other gestures related to eating, such as cutting or manipulating food, preparing foods for consumption, and resting between ingestion events. The context of these other gestures can improve the detection of periods of eating. We found that accuracy at detecting eating increased by 15% in longer windows compared to shorter windows. Overall results on CAD were 89% detection of meals with 1.7 false positives for every true positive (FP/TP), and a time weighted accuracy of 80%.

*To Adam Hoover and the Clemson Family*

# Acknowledgments

I could never acknowledge Dr. Adam Hoover's full contribution towards my growth in the last decade. That would be another dissertation. This PhD was never about the title of a PhD, or writing code as a programmer. *I wanted to be as smart as the scientists doing research.* Dr. Hoover held my hand and showed me in detail the challenges of doing good research, the responsibilities that come with it, and the political challenges you have to sometimes deal with. He transformed a lazy and undisciplined programmer to a person who understands research and can lead it. I don't think I would finish my PhD if not for his patience and dedication to supporting his students, and I am forever grateful for his approach.

I would also like to acknowledge all the love and support from the greater Clemson family - the faculty, staff and friends who have touched my life. I've spent a quarter of my life at this university, and the people around me have made it worth every minute of my time here. The faculty and staff fully supported all the crazy ideas I came up with and let me shine - from being given the charge of leading a robotics team to teaching a class of 120 sophomores, each and every experience has made me a better person.

I thank my committee members Dr. Brian Dean, Dr. Yongqiang Wang and Dr. Yingjie Lao for their time, and their interest in my research.

It would be hard to complete any research without funding, and I am thankful for support from the National Institutes of Health (NIH) grant #1R01HL118181-01A1.

A large part of the research described in this dissertation is due to a extensive data collection effort that lasted over a year. I am thankful to Dr. Eric Muth and Dr. Philip Jasper for their time and effort.

# Table of Contents

<b>Title Page</b> . . . . .	<b>i</b>
<b>Abstract</b> . . . . .	<b>ii</b>
<b>Dedication</b> . . . . .	<b>iv</b>
<b>Acknowledgments</b> . . . . .	<b>v</b>
<b>List of Tables</b> . . . . .	<b>viii</b>
<b>List of Figures</b> . . . . .	<b>ix</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 mHealth . . . . .	2
1.2 Obesity . . . . .	4
1.3 Monitoring EE . . . . .	6
1.4 Monitoring EI . . . . .	7
1.5 Previous Work . . . . .	9
1.6 Sensors . . . . .	18
1.7 Classifiers . . . . .	22
1.8 Variability in Collected Data . . . . .	26
1.9 Transitioning to Experiments in Free living . . . . .	27
1.10 Metrics for evaluation . . . . .	29
1.11 Contributions of this work . . . . .	33
<b>2 An experiment to replicate previous work on meal detection</b> . . . . .	<b>35</b>
2.1 Shimmer3 Sensor Module . . . . .	37
2.2 Linear Acceleration of the wrist . . . . .	39
2.3 Rotating Coordinate Axis . . . . .	44
2.4 Software for Data Collection . . . . .	49
2.5 Data Collection . . . . .	50
2.6 Detection of Eating from Wrist Motion Data . . . . .	51
2.7 Results . . . . .	56
2.8 Conclusion . . . . .	60
<b>3 Filtering Linear Acceleration when Tracking Wrist Motion</b> . . . . .	<b>63</b>
3.1 Methods . . . . .	64
3.2 Results . . . . .	70
3.3 Discussion . . . . .	72
<b>4 The Impact of Walking and Resting on Automated Detection of Meals</b> . . . . .	<b>73</b>
4.1 Overview . . . . .	75

4.2	Walking and Resting Detection . . . . .	76
4.3	Free-living Eating Activities . . . . .	80
4.4	Results . . . . .	85
4.5	Discussion . . . . .	90
<b>5</b>	<b>Using deep learning to detect periods of eating . . . . .</b>	<b>92</b>
5.1	Methods . . . . .	93
5.2	Results . . . . .	98
5.3	Conclusion . . . . .	102
<b>6</b>	<b>Discussion and Conclusion . . . . .</b>	<b>104</b>



# List of Tables

1.1	Common convolutional filters for image processing. . . . .	25
1.2	Complexity and size of datasets in previous works. . . . .	27
1.3	Change in precision and recall of meal detection when transitioning from a controlled environment to free-living. . . . .	28
1.4	Example of a confusion matrix for a classifier’s performance. . . . .	31
2.1	Demographic details for participants recruited for CAD . . . . .	51
2.2	Number of meals in the dataset by day of week. . . . .	51
2.3	Feature values from previous work . . . . .	59
2.4	Feature values from this work. . . . .	59
2.5	Comparison of quantity of data between previous work and this new data collection	60
2.6	Time difference between logged meal start and end times, and nearest peaks . . . . .	60
2.7	Change in accuracy in replication experiment . . . . .	60
2.8	Change in precision and recall of meal detection when transitioning from a controlled environment to free-living . . . . .	61
3.1	Change of Specificity, Sensitivity, and Accuracy (%) in the meal detection algorithm after implementing the zero mean filter . . . . .	71
4.1	Change in precision and recall of meal detection when transitioning from a controlled environment to free-living . . . . .	73
4.2	Percentage of walking and resting detected during periods of walking and eating . .	86
4.3	Some of the activities reported during a meal. . . . .	87
4.4	Confusion matrix for the Clemson All-day data set (CAD) (hours). . . . .	88
4.5	The effect of secondary activities on meal detection . . . . .	88
4.6	Time difference between logged meal start and end times . . . . .	88
5.1	Eating Episode Metrics For Select Previous Work Using Wrist Tracking Data . . . .	100
5.2	The CNN classifier performs better than the Bayesian classifier. . . . .	101
6.1	Precision and $F_1$ show a downward trend as data set sizes increase. . . . .	105

# List of Figures

1.1	The ELMM watch . . . . .	2
1.2	Examples of devices capable of recording ECG signals . . . . .	3
1.3	RileyLink connection schema . . . . .	4
1.4	Options for weight loss . . . . .	5
1.5	A room calorimeter . . . . .	6
1.6	Examples of electronic food diaries. . . . .	8
1.7	Example of a microphone sensor for detecting chewing sounds . . . . .	11
1.8	Example of a throat microphone to detect swallowing . . . . .	13
1.9	Example of a strain sensors used for detecting chewing . . . . .	13
1.10	Piezoelectric sensor used in work by Sazonov et. al to track jaw motion during chews . . . . .	14
1.11	A Wrist activity tracker connected to a laptop . . . . .	16
1.12	Proximity sensor to detect eating . . . . .	17
1.13	MPU6000 IMU manufactured by TDK. . . . .	18
1.14	Fusing IMU data to obtain orientation information . . . . .	19
1.15	Accelerometer behavior during free-fall . . . . .	19
1.16	The Coriolis effect . . . . .	20
1.17	Orientation of the Earth's magnetic field . . . . .	21
1.18	Strength and declination of the Earth's magnetic field . . . . .	22
1.19	A perceptron operating on 3 inputs . . . . .	24
1.20	Eating in free-living vs. eating in a lab . . . . .	27
1.21	Evaluating the performance of detecting eating episodes. . . . .	29
1.22	Labeling of eating time metrics . . . . .	30
2.1	iPhone4 mounted on the right arm for data collection . . . . .	36
2.2	Comparing the size of the Shimmer3 to the iPhone . . . . .	36
2.3	Photo of the Shimmer3 . . . . .	38
2.4	Screenshot of the configuration screen in Consensys. . . . .	39
2.5	Einstein's happiest thought (1907) . . . . .	40
2.6	An accelerometer in free-fall . . . . .	40
2.7	An accelerometer at rest . . . . .	41
2.8	An accelerometer resting on a surface being pushed to the right . . . . .	41
2.9	Yaw, pitch and roll Euler angles for an airplane . . . . .	43
2.10	Rotations between coordinate axes. . . . .	45
2.11	Accelerometer and gyroscope coordinate axes for the Wrist coordinate system . . . .	46
2.12	Right hand rule . . . . .	47
2.13	Data from the Shimmer has to be transformed to the wrist coordinate system. . . .	47
2.14	Screenshot of MarkerParser . . . . .	49
2.15	One day of wrist motion activity . . . . .	52
2.16	Two hours of wrist motion activity . . . . .	53

3.1	Source contributing to error in linear acceleration. . . . .	67
3.2	Magnitude of linear acceleration vs time . . . . .	68
3.3	An example of the composition of noise in linear acceleration during motion. . . . .	68
3.4	An example of the composition of noise in linear acceleration during rest . . . . .	69
3.5	Flowcharts showing the eating detection algorithm with and without filtering . . . . .	70
3.6	Spread of acceleration values showing noise and signal . . . . .	71
3.7	Linear Acceleration after filtering . . . . .	71
4.1	Sensor data while eating a banana . . . . .	75
4.2	Sensor data while walking . . . . .	75
4.3	Sensor data during a meal . . . . .	75
4.4	Sensor data while walking and eating a banana . . . . .	75
4.5	Histogram showing amount of zero-crossings per subject in the different datasets. . . . .	79
4.6	Histogram showing the amount of rest per meal in the cafeteria dataset. . . . .	79
4.7	Wrist motion activity during a full day . . . . .	82
4.8	More wrist motion data examples . . . . .	85
4.9	Five example meals from the free-living dataset . . . . .	89
5.1	Generation of continuous probability of eating during the day. . . . .	94
5.2	Detecting periods of eating using a hysteresis based segmenter. . . . .	94
5.3	Cross-folding and data preparation. . . . .	95
5.4	Neural network design . . . . .	96
5.5	Accuracy of classification vs Window size $W$ . . . . .	99
5.6	Effect of $T_S$ on TPR and FP/TP. . . . .	100
5.7	Effect of $T_E$ on average boundary error and FP/TP. . . . .	100
5.8	Histograms showing lengths of self-reported meals in CAD and the percent of meals detected . . . . .	102
6.1	Precision of detecting eating activities drops as the data set sizes increase. . . . .	105

# Chapter 1

## Introduction

This work considers the problem of detecting when humans eat by tracking wrist motion all day. Wrist motion can be tracked using accelerometers and gyroscopes. These sensors are readily available in modern wearable devices like smart watches. They measure acceleration and angular velocity, from which a multitude of features can be calculated including pose (orientation) and linear acceleration (giving a very rough measure of distance moved). Classification algorithms can then be designed to detect eating activities by looking for patterns related to hand-to-mouth gestures. The goal is to identify features and patterns that are indicative of eating compared to other gestures throughout the day. The automatic detection of eating activities can be then be used to improve daily measurements of energy intake, which is extremely important in the fight against obesity. The EMM watch (see figure 1) is a wrist worn device that can measure bite count during a meal, from which an estimate of energy intake can be derived [1]. However, it requires that the user manually turn bite counting on and off at the beginning and end of a meal. The work discussed in this document is intended to automate this process so that human intervention is not required.



Figure 1.1: The ELMM watch, one example of a wearable wrist based device that can track energy intake through bites.

## 1.1 mHealth

The last century has seen a transition in epidemiological causes of death, wherein chronic causes have now exceeded infectious and accidental causes of death [2], requiring a shift in approaches to medical care and lifestyles. The current model of medical care is reactive. Care is provided to a patient when they approach a professional for treatment, which is precipitated by a medical event, infection or injury requiring professional intervention. Evidence and observations prior to a hospital visit are often anecdotal, requiring doctors to guess details. Given the transition in epidemiology, a new approach enabling a healthy lifestyle has merits. A preventive and on-line approach to a healthy lifestyle would consist of personal health monitors objectively monitoring a subjects health. Health metrics could be monitored on-line and in real time, giving a comprehensive history of a patient's health to a medical professional.

On-line recording of medical metrics is being aided by sensors and device manufacturing processes resulting in smaller form factors. Technological advances in sensor technology and miniaturization throughout the last decade have made portability feasible. Consider the electrocardiography (ECG) machine, a device that records the electrical activity of the heart. ECGs are commonly performed to detect cardiac problems. The ECG machine displays the observed sinus rhythm of the heart. This is done by plotting the voltage difference between two leads connected to the body. An ECG machine is typically an expensive, bulky machine mounted on wheeled carts to allow for portability. Figure 1.2 shows an example of one such machine. Compare this to newly released Apple Watch Series 4 (September 2018), which is also capable of recording and displaying ECGs.



(a) ECG Machine



(b) Apple Watch

Figure 1.2: Examples of devices capable of recording ECG signals. (left) A digital ECG machine with a keyboard and display mounted on a wheeled cart with average cost of \$2000 (right) Apple Watch 4 retailing at \$429 also capable of tracking ECG signals.

The device retails for \$429 and can monitor ECG signals in real time and on-line at the press of a button, without any intervention from a professional. In the last few years, news media has multiple reports [3–5] of Apple Watch heart rate notifications leading to timely intervention in the case of a heart attack.

Another example of the democratization of medicine is an the insulin pump (artificial pancreas) used by some diabetics. A security flaw in insulin pumps manufactured by Medtronic allowed hackers to modify older models and connect them to a smartphone, using a bluetooth controlled module called the RileyLink (figure 1.3), allowing sophisticated algorithms that can control blood sugar. The current Medtronic pump costs \$7500 and cannot be customized to an individual. The FDA has also not approved the pump for children. This has led to RileyLink (\$250) devices being used by parents for toddlers dealing with diabetes [6].

Modern advancements in computing, technology and medicine are rapidly disrupting human lifestyles. Consider the simple act of hailing a taxi in New York. One would have to stand on the side walk and signal a taxi driver passing by. Ridesharing apps like Uber and Lyft have disrupted this system. A user can now request a taxi from the comfort of their home using a mobile phone, and

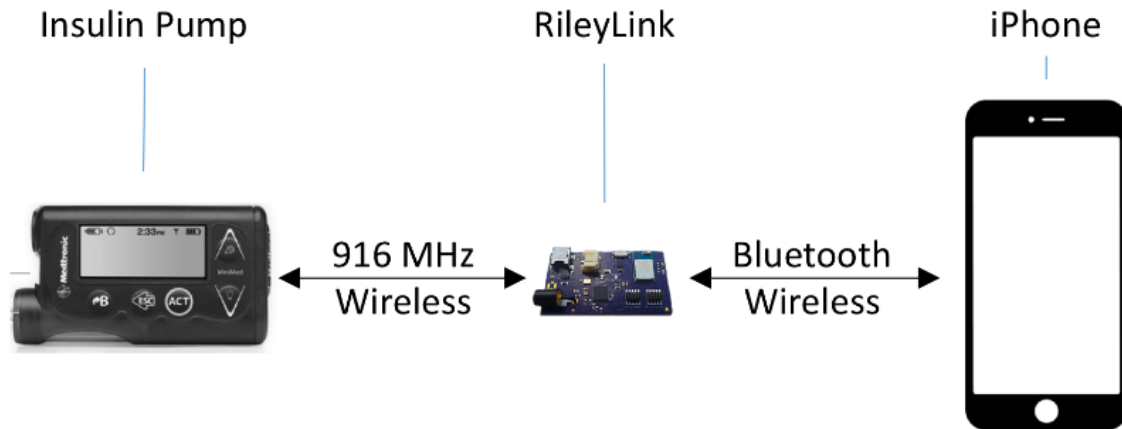


Figure 1.3: Figure from RileyLink’s website showing how a phone can connect to an insulin pump and use better control algorithms.

have it waiting for them when they need to leave. Similar disruptions have been made by Amazon in the domain of shopping (products delivered to the user), and food delivery applications.

We envision a world where health and medicine is also in the control of each user, and instead of a person visiting a medical facility when needed, we imagine a world where custom tailored medicine is delivered to them. The focus of this work is a device that can help measure daily energy intake (EI) or the food consumed by a person, assisting in the fight against obesity.

## 1.2 Obesity

The CDC reports that obesity was prevalent in 39.8% of the population and affected about 93.3 million of US adults in 2015-2016 [7]. This condition has been shown to highly correlate with serious diseases like diabetes, stroke heart disease and certain types of cancer, some of the leading causes of preventable, premature death today [8,9]. While research on obesity is ongoing, its contributing factors have been identified as behaviors and genetics. Behaviors can include dietary patterns, physical activity, inactivity, medication use, and other exposures [10]. Put simply, this can be considered a problem of a trending imbalance in energy intake (EI) and energy expenditure (EE). If EI is more than EE, food is converted to fats and stored by the body - a mechanism in human physiology stemming from food scarcity. This leads to a somewhat trivial “solution” to weight loss - consuming less energy than being expended by the body. However, this is very hard in practice given trends in dining behavior and environmental settings that promote excessive food intake and



Figure 1.4: Options for weight loss include surgeries and lifestyle changes like more physical activity, and reduced energy intake. [12,13]

discourage physical activity [11].

Weight loss can be achieved through various means, for example surgical procedures like liposuction, abdominoplasty or bariatric surgery, or lifestyle modification such as reducing energy intake or increased physical exercise. However, not all methods for weight loss are equally effective. For example, surgical methods like liposuction or abdominoplasty involve the removal of fat or mass from the body, which leads to weight loss. However the mass removed is often minor, and the effect of these surgeries is merely cosmetic as they do not affect eating habits or physiological responses of the human body to food [14]. For sustained weight loss, subjects must modify their lifestyles to create an imbalance in energy intake and expenditure [15]. Another invasive method, bariatric surgery is thus more effective than liposuction alone, as in this method the stomach is surgically operated and reduced in size. This leads to less food being consumed as subjects feel satiated faster, affecting the physiological response of the body. While bariatric surgery is efficient, a large majority of the population prefers weight management through diet change and increased physical activity as surgery is invasive and expensive [16]. Lifestyle modification can be achieved through increased physical activity (EE) and decreased energy intake (EI).





Figure 1.5: Inside a room calorimeter [17]

### 1.3 Monitoring EE

The field of monitoring energy expenditure includes many methods. The gold standard is the calorimetric chamber (figure 1.5) - an entire room constructed just for the purpose of estimating energy expenditure. These rooms are designed to measure  $O_2$  and  $CO_2$  concentrations released by a subject. A subject has to inhabit this chamber for extended periods of time. By measuring changes in the concentrations of these gases, an estimate of the energy expended can be obtained. While accurate, these methods are not feasible for use in free-living humans, where other more convenient methods are wanting. Self-monitoring is an option, where subjects track physical activity in a diary. Estimates of energy expenditure can be made from the type and duration of the physical activity, however this process is burdensome. This has led to the creation of automatic physical activity tracking. Physical activity like walking or running [18] can be tracked using accelerometer and gyroscope data. Fitbit's physical activity trackers are well known devices that track such activity. By enabling automatic logging of steps, distance walked and other physical activities, the Fitbit app lets users track their metrics, like energy burned. Additional features allows a user to set a goal for weight gain or weight loss, or log meals in an electronic food diary. The app then suggests a routine or daily step count goal, and monitors progress towards this goal. The ease of use and the

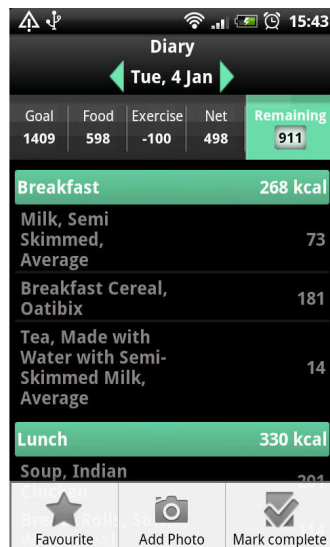
desire to maintain healthy lifestyles is helping with the public acceptance of personal monitors like the Fitbit [19] that promote well-being by constantly monitoring personal metrics.

While increasing EE is a component of many fitness routines, decreasing energy intake has been shown to be more effective in controlling obesity than increasing its expenditure, as physical activity often does not contribute considerably to large amounts of energy expenditure or weight loss [20].

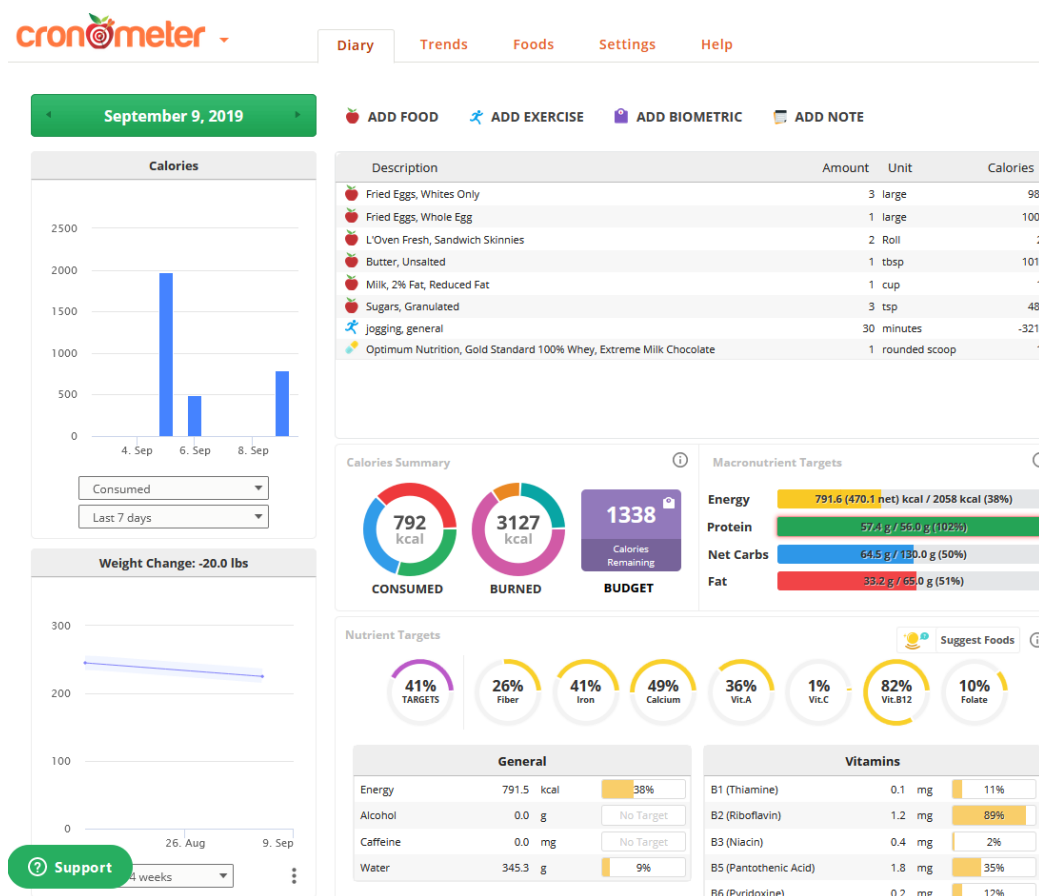
## 1.4 Monitoring EI

It has been shown that monitoring energy intake assists in reducing it [21,22]. Energy intake can be tracked through tools like food diaries and calorie tables, or with assistance from trained dietitians who estimate energy intake for subjects using meal recall or food frequency questionnaires [23]. Doubly labeled water (DLW) [24,25] is considered the gold standard for energy intake estimates. When using DLW, the atoms of water are replaced by an uncommon isotope, which enables the tracing of these elements. A subject is administered with a dose of DLW, and then elimination rates of the isotope are measured, which helps estimate the daily metabolic rate. However this method is complicated and slow - usually taking 7 to 14 days [26].

Self-monitoring is preferred to DLW due to its ease of use. Individuals can note their eating activities and the foods they eat in a food diary. Energy estimates can then be made by the individual themselves or by an expert. However this has shown to be error prone and susceptible to bias, leading to underreporting of energy intake [27–29]. For self-monitoring methods to be widely adopted, they need to be inexpensive and easy to use [30]. This underreporting and bias might be addressed by adding an unbiased agent, such as a computer algorithm. The increase of computer and smart phone usage has led to attempts in digitizing such tools (figure 1.6), for example a website based food diary [31], or smartphone applications that let subjects use the phone as a food diary [32]. These implementations allow users to pick food items from a database, and automatically estimate calories consumed using in-built nutrition information, thus somewhat reducing the burden on a subject to log this information. Smartphone based logging has shown higher adherence than (93%) compared to a paper diary (53%) after 6 months, as self-monitoring burden causes a drop in usage and adherence over time [32]. However, sustained self-monitoring of energy intake remains a challenge. Helander et. al retrospectively analyzed data from ~190,000 users of the food diary app



(a) Phone Application - MyFitnessPal



(b) Website - cronometer

Figure 1.6: Examples of electronic food diaries.

Eatery.com, and learned that less than 3% of users continued using the app more than 1 week [33].

It would help subjects tracking EI if EI could be measured automatically using wearable devices, just like energy expenditure can be. Researchers in the recent field of Automated Dietary Monitoring (ADM) aim to unobtrusively and automatically monitor energy intake using electronics and sensors instead of human intervention. The goal of ADM is to better estimate energy intake by way of reducing the burden of self-monitoring, and reduced user bias.

The motivation for this work was partly the ELMM Watch, which can track the number of hand to mouth gestures (“bites”) performed by a user. The device currently requires a user to press a button at the start (and end) of a meal to indicate that they are consuming food. When not eating, this watch triggers spurious false positives, caused by motions that look similar to consuming a bite of food. Users often forget to indicate the start or end of a meal, leading to meals not being tracked or spurious gestures being included as bites [1, 34]. A previous research showed that of all the records collected by the ELMM watch in a 2 week trial at Clemson University, 9% were invalid, while during longer (16 and 26 week) studies, the number of invalid records increased to 34% [35]. These invalid records were likely caused to to fatigue with remembering to mark the start or end of a meal on the ELMM watch. The proposed meal detection algorithm would automatically detect if food was being consumed or not without user input, thus improving accuracy of a device like the ELMM watch.

## 1.5 Previous Work

Food is consumed through a series of actions that can be tracked using various sensors. Sensor data from these actions can be used to detect eating. For example, human subjects typically acquire food, which may be placed on a table or flat surface. This might trigger a short walk and movements that can be detected using accelerometers. With the food ready for consumption, the subject might first perform hand gestures like manipulating the food or using condiments. When ready to consume the food, the hand brings the food to the mouth, and then returns to the resting position, or begins manipulation for the next bite. These hand gestures are typically tracked using accelerometers or video cameras.

Once the food enters the mouth, chewing begins, which creates sounds and jaw movement. These sounds can be tracked using a microphone, while jaw movements can be detected using strain

sensors. Once food has been sufficiently broken down by mastication, swallowing occurs, where the tongue pushes the food towards the esophagus. These movements cause physical movements that can be detected by Piezoelectric or strain sensors, electrical activity in the muscles that can be tracked by Electromyography (EMG) sensors, and sounds that can be picked up by microphones.

By detecting multiple microevents or microgestures, and then combining them together, eating activities can be detected. Various groups have attempted different modalities and methods to detect eating. This section reviews some of the work done previously by other researchers.

### 1.5.1 Detecting Chewing

Early work in ADM was based on detecting the sounds of chewing [36], due to the ease of tracking such sounds - all you need is a microphone, and the presence of chewing sounds when food is consumed. Amft et. al showed that the sounds of chewing can be recorded by a microphone mounted inside the ear canal and then processed to detect chews and chewing cycles [36]. Furthermore, foods that have different textures (eg. hard, soft, chewy, crispy) can also be identified due to the difference in their chewing sounds [36]. In this work, the authors reason that a microphone mounted inside an ear canal is unobtrusive given their widespread use in applications like hearing aids and headsets. Four participants were recruited to consume chips, chocolate, an apple and carrots. While individual chews could identify the type of food with an accuracy of 66 - 86%, cycles of multiple chews improved the classification accuracy to 80 - 100% [36].

Later work by the group used microphones mounted behind the ear only. This is perhaps because the authors felt the IMU and EMG methods did not perform better than sound in their experiments. The group has also shown that food can be grouped into three classes based on the type of food and sounds created during chewing (wet and loud, dry and loud, and soft and quiet) [37]. The authors suggested that some foods in the soft and quiet group would perform poorly using this method due to the lack of sounds created when chewing them. The work also compares different placements of the microphone like the inner ear, near the mouth, the cheek, collar or behind the ear, and states that the inner ear or behind the ear locations are the most inconspicuous ones for users.

Sazonov and Fontana considered using piezoelectric film strain sensors to track chewing (figure 1.10). In work published in 2008, Sazonov et. al discuss collecting a large data set using multiple sensors like microphones, strain sensors and piezoelectric sensors [38]. The authors used feature selection on strain sensor data to identify 11 features they considered most critical for food

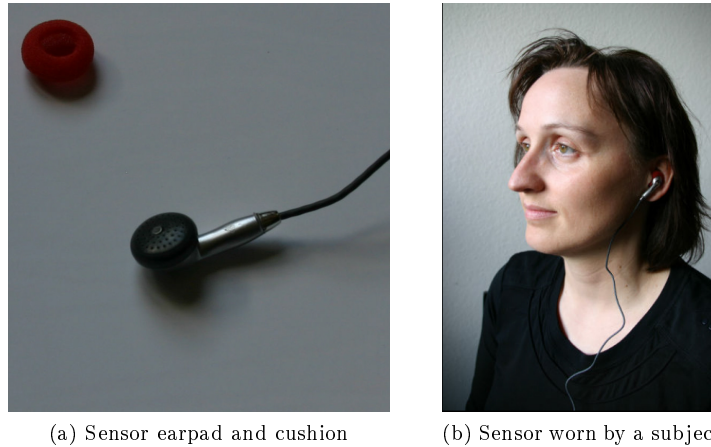


Figure 1.7: Earpad microphone design by Amft [37]

intake detection. Using support vector machines, they were able to classify epochs with an accuracy of 81% [39].

The authors collected more data using the same piezoelectric film strain sensor, since the previous data collection was small and did not contain much variance in terms of subjects and behavior. Seven subjects performed 10 minute activities like walking, talking, eating and resting. The data was segmented into epochs, for which 29 time and frequency domain features were calculated. Each epoch was represented by its features, as well as the epoch before and after it. By applying a linear support vector machine (SVM) to these feature values, the authors were able to achieve an average accuracy of 90.52% [40], showing that piezoelectric sensors can be used to detect chewing.

Both these works were built and evaluated on data collected from subjects in the lab. To work with subjects in free-living, more data was collected using the same sensor from free-living subjects [41]. Data was collected from 12 subjects recording data for 24 hours each. The jaw motion signal was reduced to 38 time and frequency domain features calculated for 30 second epochs. The authors trained a single hidden layer based neural network and a linear SVM, which were tested separately on 48 hours of data collected from a single individual. Of 11 meals consumed by the individual, the neural network detected eight (one false positive (FP)), while the SVM detected seven (three FPs). The authors learned that when employing piezoelectric sensors, excessive physical activity or exercise can often confuse classifiers and look like food intake.

Other researchers have also demonstrated methods and sensor modalities to detect chewing. Passler et al. collected data from 51 participants who consumed seven types of food and a drink.

The subjects were fitted with two microphones in a hearing aid package. By identifying chewing and swallowing sounds, the authors showed the ability to detect the kind of food being consumed [42]. Papapanagiotou et al. considered wearable photoplethysmography (PPG) sensors, which detect changes in the the volume of blood using optics. From 10 hours of data recorded from 21 participants wearing an earphone like PPG sensor, the authors showed the ability to detect chews with 70% precision and 76% using different algorithms [43]. More recently, Chun et al. demonstrated a method to detect periods of eating using a necklace like proximity sensor (figure 1.12). The method works by first detecting chewing, then classifying a period containing bouts of chewing as eating. This work is discussed in detail in section 1.5.4. In recent work, Zhang et al. have employed the use of EMG sensors mounted in eye glasses, showing another sensor modality that can be used to detect chewing using wearable devices [44].

### 1.5.2 Detecting swallowing

Amft also worked on detecting swallowing. He collected data from six participants wearing a sensor collar consisting of an EMG sensor and a microphone. This configuration was used to detect the sounds and muscle movements of swallowing [45]. Participants were requested to swallow the food items in one piece. The study was inconclusive with the attempted method performing well on data from two participants, but not on the remaining four.

In the same year Sazonov et. al collected a large set of data using multiple sensors like microphones, strain sensors and piezoelectric sensors [38]. The group planned to use microphones to track sounds and detect swallowing, the strain and piezoelectric sensors to track jaw movements and detect chewing. 20 volunteers completed four visits in a lab where they were inactive for 20 minutes (10 silent and 10 reading aloud), consumed food for 20 minutes, and then inactive (10 silent and 10 talking) for another 20 minutes. A camcorder was used to record video of the visits that was later used to provide ground truth for classifier evaluation. Subjects were also provided with a push button to press whenever they swallowed food. Background noise was simulated in two of the four visits.

After testing various configurations, they concluded that a microphone placed at the throat (figure 1.8) to detect sounds of swallowing is least susceptible to ambient noise. A piezoelectric film sensor attached below the ear using medical tape was selected (figure 1.10) to track jaw motion, as testing showed that strain sensors (figure 1.9) were more sensitive to head tilting than frequently



Figure 1.8: Throat microphone configuration used to detect swallowing in work by Sazonov et. al [38]

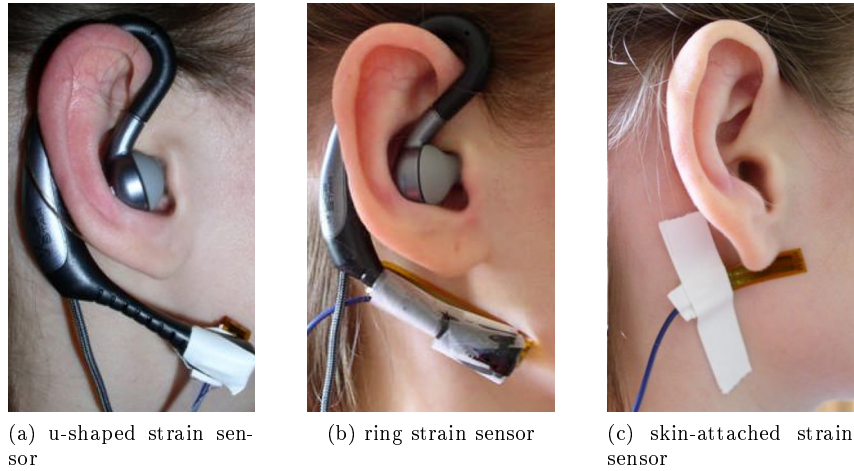


Figure 1.9: Strain sensors tested by Sazonov et. al to detect chews [38].

occurred during drinking.

Over the four visits, participants consumed two sizes of meals comprising of pizza, yogurt, an apple and a peanut butter sandwich, to represent the variance in food physical properties like crispiness, softness/hardness and tackiness. Human raters marked intake and swallow events to indicate ground truth. By considering both swallowing and chewing, along with the counts and frequencies of these events, new work was able to detect periods of food intake with  $>95\%$  accuracy, differentiate solids from liquids with  $>91\%$  accuracy, and also predict the mass of ingested food with  $>91\%$  accuracy for solid food, and  $>83\%$  for liquids [46].

In further work, features were derived using the time-frequency decomposition of sound



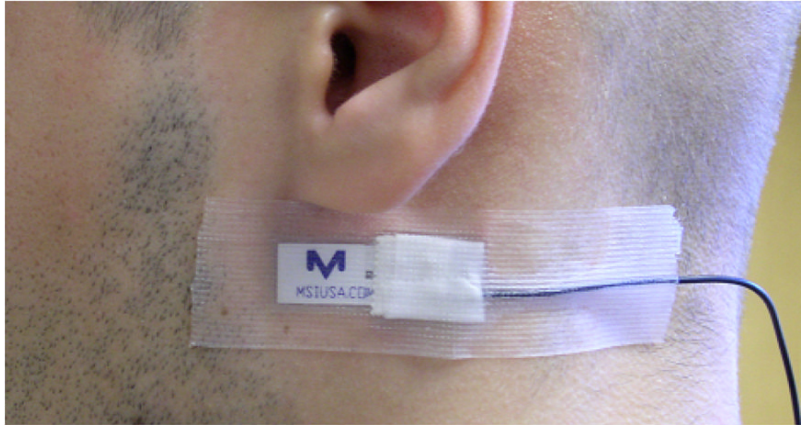


Figure 1.10: Piezoelectric sensor used in work by Sazonov et. al to track jaw motion during chews [38].

segments. When classified by SVMs, the group was able to detect swallowing events with a weighted accuracy of 84.7% [47], showing that throat based microphones can be analyzed to separate periods of food intake from others like respiration, speech, head movement or just ambient noise.

By using unsupervised learning using K-means clustering that accounts for individual traits in a person, Lopez-Meyer showed that classification accuracy could be further improved to 94% [48]. Data from the strain sensors was not used in this study, and the authors learned that sound based tracking of swallowing is sufficient for detection.

Olubango and Ghovanloo demonstrated a real time swallowing detector in 2014 [49]. Data was collected from four subjects using a throat microphone. Participants coughed, swallowed various food and drinks, and spoke. The authors showed that their method could discriminate between swallowing and speaking.

### 1.5.3 Wrist tracking to detect bites

The wrist performs specific gestures when food is brought to the mouth, sometimes referred to as hand-to-mouth (HTM) gestures, often more simply referred to as bites. Given the acceptability of watches, a number of groups have attempted using wrist based sensors to detect eating. This method often relies on instrumenting either the dominant hand or both the hands of a subject.

A recent survey by Kalantarian et. al of 221 participants showed that participants are generally supportive of wearing wrist watches on the dominant hand [50]. Of all participants, 76% preferred wearing a watch on the left hand, while 19% preferred the right hand, and 5% had no

preference. 23% of the participants stated they prefer not wearing a watch, 33% of the participants stated they always wore a watch, while the others stated they did not mind wearing a watch or would like to wear one. Of the 221 participants, 28% said they were not okay wearing the watch on the opposite hand, while the rest would consider it.

Dong et. al first demonstrated that wrist based tracking could be used to count the number of bites of food consumed during a meal [51]. The group discovered that during eating, the wrist undergoes characteristic rolling of the wrist to take a bite of food. Ten subjects were fit with an InertiaCube3 sensor on their dominant hand which tracked angular velocities of the wrist (roll, pitch and yaw). The subjects consumed a meal of their choice while wrist motion was tracked, while being recorded on video by a camera. The video was reviewed by a researcher to provide ground truth evidence of the bites. The authors reported 91% sensitivity. They also noted that the device would often pick up false positives from behavior where the hand was moved to the mouth but a bite was not taken, like using a napkin. In later work, the group collected more data. In three experiments collecting data from 102 subjects over 242 meals, the group demonstrated that the method was reliable in laboratory as well as uncontrolled environments [52].

Using data collected from a cafeteria by the same group, Ramos et. al analyzed a wrist motion dataset containing data from 25 meals consumed in a cafeteria [53]. The authors note that specific wrist gestures are often conducted during a meal, such as sipping a drink, using a utensil, or consuming a bite of food. Further, the authors note that individual gestures has sequential dependence, which can be modeled by classifiers. Using Hidden Markov Models (HMMs) to capture the inter-gesture sequential dependencies, the authors showed that it is possible to recognize individual gestures with high accuracy (97%).

In later work, Shen et. al analyzed wrist motion data collected from 271 meals in an unrestricted cafeteria setting [54]. The work discusses how variations in food type affects total wrist motion during a meal, hypothesizing the cause to be an increase in head-to-plate movements instead of hand-to-mouth movements. The authors show that the method of tracking wrist movements to detect bites is reliable, and provides an automated measure of tracking intake during unrestricted eating.



Figure 1.11: Wrist activity tracker connected to a laptop carried in a backpack [55].

### 1.5.4 Detecting Eating Events

Being able to detect chewing with satisfactory accuracy, Amft also considered detecting eating events by chewing [45]. To collect data, four participants were fitted with inertial measurement units (IMUs) on the lower arm, upper arm and upper back. The IMUs were used to detect movement during eating and other non-eating activities (scratching head, touching chin, reading a newspaper, using tissue, glancing a watch, and answering a mobile phone). Tests showed that the upper arm and back sensors did not provide enough data to aid recognition of activities, possibly due to lack of motion in this region. The results showed a recognition accuracy of 75 - 82% for the different activities conducted, using lower arm IMUs only [45].

In a second experiment, an ear microphone was used to detect the sounds of chewing. Chewing sounds were detected from a single participant consuming food [45]. While results showed that chewing sounds are not a reliable method to detect eating if soft foods are consumed, the authors showed class-relative recognition accuracy of 85 - 87% [45].

Similarly, Dong et. al built on their method of detecting bites using wrist mounted IMUs, and considered detecting eating activities using the same modality [55]. While most groups during the period collected data from subjects in controlled environments, the group collected wrist tracking data from unsupervised free-living subjects so that variance due to free-living behavior can be accounted for. Each subject was fit with an InertiaCube3 mounted at the wrist that was connected to a laptop to log data (figure 1.11). The laptop was carried in a backpack when needed. Subjects were asked to remove the device when needed, for eg. when showering or playing contact sports. Subjects recorded activity behaviors in a log book with time stamps. A reviewer classified activities



Figure 1.12: Proximity sensor employed by [57] to detect periods of eating from jawbone movement.

into three classes: eating, ambulatory and sedentary. The authors performed two experiments, in the first, a rule based classifier demonstrated 91% accuracy in assigning these classes to a total of 125 tasks [55]. In a second experiment, the authors use a state machine to categorize time into one of four categories: start of eating, end of eating, not-eating and inside or outside eating. In this experiment, the authors were able to detect eating activities with a sensitivity of 82%, and correctly identify eating activity boundaries within 10 minutes of ground truth [55].

Utilizing the lessons learned from this work, the group collected even more data. Data was collected from 43 free-living humans, wearing an iPhone on their wrist all day [56]. The iPhone4 was mounted on participants' wrist, and logged wrist accelerations and angular velocities all day. Meals were logged in two ways. Some subjects recorded eating activities using a log book, while others were able to use an updated application to mark them on the phone itself. The group showed that meals are often preceded and succeeded by periods of high wrist motion activity, and hypothesized this was caused due to activities like meal preparation or walking before a meal, and clean up activities after a meal [56]. Using these peaks to segment periods of time, the group engineered four features that captured the difference between eating and non-eating activities. Periods of time were classified using a Naive Bayesian classifier and the four features. The group demonstrated that this method to detect eating by tracking wrist motion achieves 81% accuracy on their dataset collected from 43 free-living humans [56]. This work is the primary inspiration for this thesis and is discussed in detail in section 2.6.

More recently, Chun et al. showed that periods of eating can be detected using a necklace like proximity sensor (figure 1.12). Data was collected from 20 subjects in laboratory, 15 subjects in a controlled field study, and 19 participants in free-living [57]. The authors showed a precision of

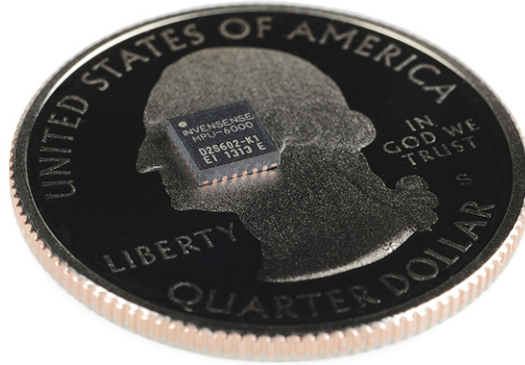


Figure 1.13: MPU6000 IMU manufactured by TDK.

91% and recall of 93% in the laboratory data, while for the free-living data the precision and recall were 78% and 73%. The authors learned that eating episode detection performance decreases as duration of segment decreases, as there are less chews for the classifier to detecting eating activity from [57]. Zhang et al. have also demonstrated a similar method of using a necklace to detect eating [58].

## 1.6 Sensors

Motion can be defined by an object’s location, orientation, displacement, velocity and acceleration. Of these units, acceleration, angular velocity and orientation can be sensed by inexpensive, commercially mass produced accelerometers, gyroscopes and magnetometers. These sensors are typically packaged in a single chip which is produced for use in a wide range of consumer electronics, especially smartphones, tablets, handheld videogames and smartwatches. An example of this is the MPU-6000 produced by InvenSense [59]. Shown in figure 1.13, the package contains an accelerometer and a gyroscope. A similarly sized chip, the MPU-9250 [60] contains an accelerometer, a gyroscope and additionally a magnetometer.

Packages like the MPU-6000 containing an accelerometer and gyroscope are referred to as inertial measurement units (IMUs) in the literature and are commonly used to make crude estimates

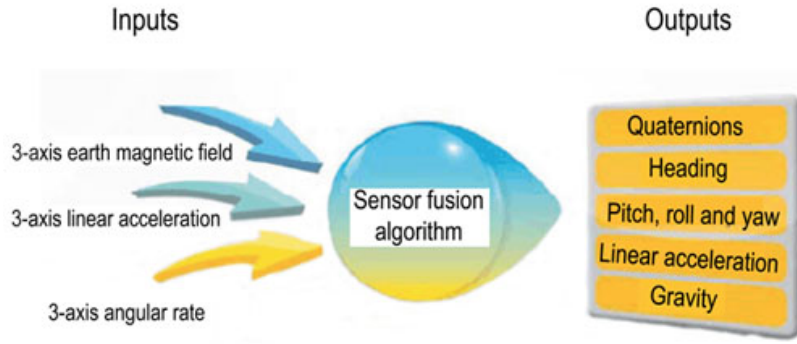


Figure 1.14: Fusing IMU data to obtain orientation information [61]

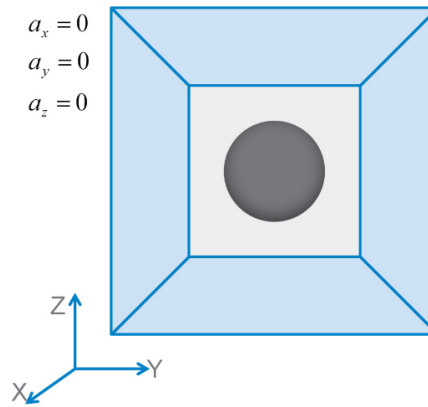


Figure 1.15: Accelerometer behavior during free-fall [62]

on the direction of motion as well as the orientation of the device with respect to the earth. Packages which also contain a magnetometer can additionally provide orientation information with respect to the earth by sensing the earth's magnetic field. These packages are often called MARG (magnetism, angular rate and gravity) sensors. Using data from MARG sensors, and sensor fusion algorithms that combine this data, orientation, gravity, and other parameters can be obtained (figure 1.14).

Work in [56] used linear acceleration and gyroscope data from an iPhone4 mounted on the subjects wrist. For our work, data was collected using the Shimmer3 wearable sensor platform which houses an MPU-9250 MARG sensor. This section briefly discusses accelerometers, gyroscopes, magnetometers and the Shimmer3.

### 1.6.1 Accelerometers

Accelerometers are sensors that sense acceleration by measuring the force exerted on an

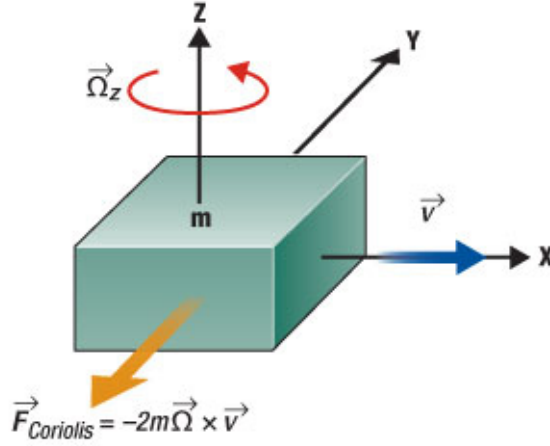


Figure 1.16: The Coriolis effect [63]

internal sensor mechanism. Due to this they are not capable of separating the effect of a gravitational field from linear acceleration. Due to this, accelerometers sense a combination of the gravity vector and force applied onto an object. This behavior of accelerometers implies that these devices can be used to estimate orientation or gravity if it is known that the device is at rest. However, additional processing must be done if a user desires to obtain linear acceleration, as gravity must first be estimated from orientation estimates, and then subtracted from the values reported by the accelerometer. Accelerometer behavior and data processing are discussed in detail in section 2.2.

### 1.6.2 Gyroscopes

Gyroscopes measure angular rate (angular velocity), which is the rate at which a device is being rotated. These sensors can thus be used to ascertain how fast a wrist is rotating and help with activity classification, for example the rolling of the wrist during a punch is much faster than during a bite of food, while there is very little rotation of the wrist when brushing the teeth. Angular velocity can also be used to estimate orientation by integrating over time. They are commonly used to improve orientation estimates in conjunction with data from accelerometers and magnetometers, since noise in sensor data from gyroscopes causes drift in estimates made using dead reckoning methods. Gyroscopes use a capacitance based method to estimate Coriolis effect, which is linearly proportional to the angular velocity of the device.

Figure 1.16 adapted from [63] shows the coriolis force experienced by an object moving at velocity  $V$  that is rotated with an angular rate  $\omega$ . MEMS gyroscopes are manufactured using

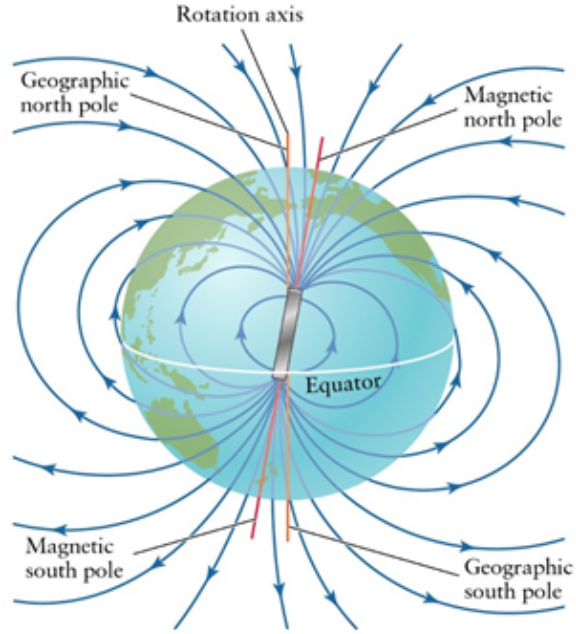


Figure 1.17: Orientation of the Earth's magnetic field [64]

the tuning fork configuration, in which two masses oscillate in opposite directions. Rotation causes equal Coriolis forces to act on both the masses in opposite directions, thus resulting in a capacitance change which is proportional to the angular velocity. Only linear velocity does not generate a Coriolis force, thus gyroscopes are capable of isolating rotational movement from linear and reporting it.

### 1.6.3 Magnetometers

Magnetometers are sensors that detect magnetic fields. In the absence of man made magnetic fields, a magnetometer can be used to identify earth's magnetic north and south pole (figure 1.17). By comparing against a database of the earth's magnetic field strength and declination, magnetometers are used as compasses in mobile phones and other devices (figure 1.18). Magnetometers alone cannot provide enough information to ascertain orientation, so their most common use is in orientation sensor fusion methods (figure 1.14). Combined with data from accelerometers and gyroscopes using sensor fusion algorithms, magnetometers provide full orientation information.

Production of these sensors was enabled by the discovery of the Hall effect - the production of a potential difference across an electric current through which a current is flowing, in the presence of a magnetic field. Modern magnetometers built using MEMS technologies may either use a



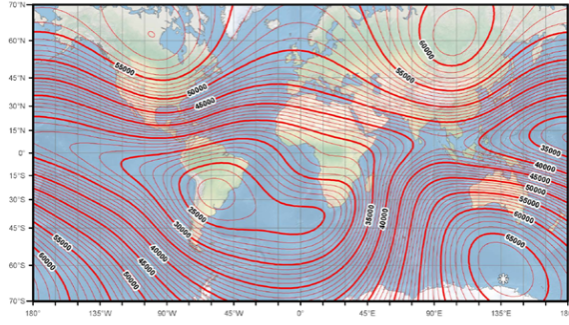


Figure 1.18: Strength and declination of the Earth's magnetic field [64]

semiconductor based hall effect sensor, or a small magnet that changes position in the presence of a magnetic field, the effect of which can be detected and calibrated to provide orientation. The end result is a sensor that is capable of providing heading - directions to the magnetic north pole of the earth.

## 1.7 Classifiers

The problem of detecting periods of eating is often formulated as a classification problem. Given certain features, a procedure assigns a label or a class to a sample. In the domain of detecting periods of eating, having ground truth information - if a subject was eating or not eating, we can first segment the data into periods of time, and then assign it a ground truth label. A set of features can be calculated for each segment - for example, amplitudes or frequencies of sounds from chewing, voltages from strain gauges or accelerations from wrist movement. Given this feature set and ground truth label, a classifier can be trained to identify time segments belonging to an unknown class - eating or non-eating. This section provides background on classifiers used in this work: The naive Bayesian classifier and neural networks.

### 1.7.1 Naive Bayesian Classifier

The naive Bayesian classifier belongs to the family of probabilistic classifiers. These classifiers model the probability of features belonging to a given class, and then use this information to predict the class for a new sample. The Naive Bayesian classifier based on the Bayes theorem is the most widely used in this family. Given a sample represented by the feature  $f$ , the Naive Bayesian classifier assigns probabilities  $p(C_i | x)$  for each possible class  $C_i \in C$  using Bayes theorem, shown in

equation 1.1.

$$p(C_i | x) = \frac{p(C_i) p(f | C_i)}{p(f)} \quad (1.1)$$

where  $p(C_i)$  represents the prior probability of the class: how often the does this class occur in the dataset,  $p(f | C_i)$  represents the likelihood of feature value  $f$  in class  $C_i$ , and  $p(f)$  represents the probability of feature value  $f$  occurring in the dataset.

Given that  $p(x)$  does not depend on  $C$ , equation 1.1 can be reduced to:

$$p(C_i | x) = p(C_i) p(x | C_i) \quad (1.2)$$

The argmax operator can then be applied to all possible class probabilities  $p(C_i | \mathbf{x})$  to obtain the class label:

$$c_i = \arg \max_c P(c_i) P(f|c_i) \quad (1.3)$$

For a problem with multiple features, this equation can be written as:

$$c_i = \arg \max_c P(c_i) \prod_j P(f_j|c_i) \quad (1.4)$$

where features can range from  $j = 0$  to  $N$ .  $p(x | C_i)$  is often modeled using a normal distribution. After splitting the training data into the ground truth classes, mean ( $\mu_i$ ) and variance ( $\sigma_i^2$ ) for the features in each class is estimated. The Gaussian probability for a normal distribution given the mean and variance is then expressed as:

$$P(f|c_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(f - \mu_i)^2}{2\sigma_i^2}\right) \quad (1.5)$$

The Gaussian naive Bayes classifier has two shortfalls: it assumes that all features are independent, and that features are distributed normally. These assumptions are often incorrect in data collected in natural world settings. However, the classifier has shown to be effective in many real world scenarios [65]. Developing the model and training the classifier is computationally inexpensive, and thus often preferred to other classifiers like support vector machines or neural networks.

### 1.7.2 Neural Networks

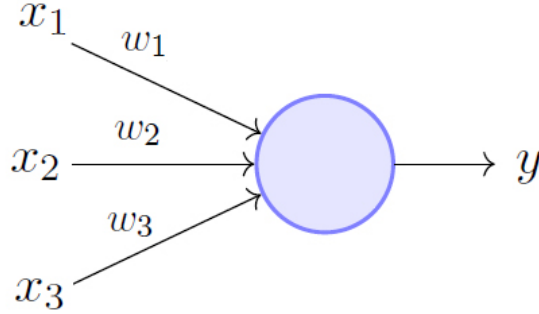


Figure 1.19: A perceptron operating on 3 inputs [66]

Neural networks are a family of classifiers that use numerical weights to calculate features from input data, and then obtain an output. Neural networks are called so because their model was inspired by the connections of neurons in animal brains. Marvin Minsky and Seymour Papert [67] first introduced a single node unit called the perceptron (fig 1.19), which can be mathematically written as equation 1.6.

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \quad (1.6)$$

where  $\mathbf{x}$  is a vector of all the inputs to the perceptron,  $\mathbf{w}$  is a vector of weights, and  $b$  is a bias.

Binary classification using the perceptron can be achieved by using a simple threshold, i.e.

$$y = \begin{cases} 1 & \text{if } f(\mathbf{x}) > 0, \\ 0 & \text{otherwise} \end{cases} \quad (1.7)$$

Given a training set  $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_s, y_s)\}$  containing  $n$  samples with features  $\mathbf{x}$  and ground truth labels  $y$ , a gradient descent algorithm can be applied to train the weights in  $\mathbf{w}$ .

The drawback of the perceptron is that a single perceptron is only able to classify linearly separable data. When introducing the perceptron, Minsky and Papert mathematically showed that a single perceptron would fail to model non-linear functions, such as the exclusive-or gate [67]. The authors also stated another issue: computational power at the time was not enough to train multiple layer perceptron models, leading to stagnation of research in the field.

Interest in neural networks was renewed later due to an improvement in computational tech-

niques on two fronts: Faster computing hardware, and parallelization techniques in software. Since gradient descent algorithms could be parallelized, this meant large improvements in the time required for gradient descent algorithms to converge, making it feasible to train a multilayer perceptron.

Over the last decade, many groups have demonstrated neural network architectures that perform better over other classifiers. This has led to an increasing amount of interest from researchers. Traditionally, multilayer perceptron networks were prone to overfitting, and thus improvements have been shown in network architectures that allow the network to generalize to the sample data better. Other improvements have been shown that improve the gradient descent algorithm to converge faster thus requiring lesser training time.

Today, neural network architectures are plenty and their designs enable networks to learn non-linear relationships. They can be summarized as a collection of layers that manipulate the input vector to an output vector, which can then be used for classification or regression. Layers in neural network architectures can be broadly classified into three types: dense, convolutional, and recurrent.

### 1.7.2.1 Dense

A dense layer, also known as a fully connected layer is the most basic layer in a neural network, and can be explained as a layer containing multiple perceptrons. The layer allows a many to many operation on its inputs. These layers are generally followed by an activation function that clips the outputs to a smaller range. While the easiest to design and train, they only operate on inputs linearly, and lack the capabilities of convolutional layers, which can automatically segment data or learn local relationships in the input vectors or recurrent layers, which can learn patterns in samples over time.

### 1.7.2.2 Convolutional Layer

Table 1.1: Common convolutional filters for image processing.

Edge Detection

-1	0	1
-2	0	2
-1	0	1

Smoothing

0	-1	0
-1	5	-1
0	-1	0

Blurring

$\frac{1}{16} \times$				1	2	1
				2	4	2
				1	2	1

Convolutional layers were inspired by work in computer vision where convolutional filters were commonly used for image processing. By convolving filters across an image, operations such as

blurring, sharpening or edge detection can be implemented. Table 1.1 lists example 3X3 filters for these operations. Weights for these filters were arrived at through hand engineering. Convolutional layers perform similar convolutional operations using filters, except the weights for these filters are learned using gradient descent.

### 1.7.2.3 Recurrent Layer Neural Networks

Dense and convolutional layers consider each sample in the dataset independently and classify it independently. This approach might not be suitable for time series data where segments of data might be dependent on each other, or have a transition probability that can be used to model the behavior of input data. For example, there is a high probability that after manipulating food the next gesture performed will be that of a bite [53, 54]. Similar to hidden Markov models, recurrent neural networks are able to model this sequence [68]. Compared to HMMs, recurrent layers learn features as the weights of a layer using gradient descent [68]. While the dense or convolutional layers process one sample as input, to produce an output, recurrent layers instead consider a sequence of samples. Each recurrent layer generates output by considering the input from the current sample as well as the output of processing neighboring samples. However, recurrent layers can often perform poorly and not converge when the input samples are long [69]. Variants of recurrent layers, like the gated recurrent Unit (GRU), or long-short term memory organize weights into specific roles to better model long sequences [69].

## 1.8 Variability in Collected Data

Table 1.2 lists free-living datasets collected by previous works. These datasets contain sensor data from periods of eating and non-eating collected from 7 - 43 participants. Such limited sizes of participant pools may not capture the behavior variability observed in the general population. In addition, the ratio of data collected in the eating class and non-eating class is not standard, with different works collecting data in different ratios (3:1 to 20:1). An average person spends 1.17 hours a day in activities relating to eating [70], a non-eating to eating ratio of appx. 20:1. As discussed later in section 1.10 this ratio affects metrics like precision and the F1 score, making it hard to compare works. Such sample sizes (7 - 43) are often sufficient in research with deterministic methods such as in physics, chemistry or engineering. However, psychological and behavioral studies are often

Table 1.2: Complexity and size of datasets in previous works.

Authors	Hours	Subjects	Ratio
Dong et. al [56]	449	43	20:1
Thomaz et. al [71]	32	7	14:1
Bedri et. al [72]	45	15	8:1
Bedri et. al [72]	12.2	10	3:1
Mirtchouk et. al [73]	245	11	12:1
Zhang et. al [74]	122	10	17:1
Farooq et. al [75]	10	40	16:1

complicated and find variance from one individual to another, thus often recruit a larger range of participants (48 - 411) for studies [76, 77]. In work from 2013, Button et al [78] discuss why a small sample size undermines the reliability of neuroscience studies, mentioning how reproducibility, a key priority of research, is hindered. By collecting data from more subjects, more variation in behavior can be observed. This may in turn also reduce classifier accuracy. Chapter 2 discusses this further. By collecting an order of magnitude more data than previous work, we quantify how the number of subjects affects classifier performance.

## 1.9 Transitioning to Experiments in Free living

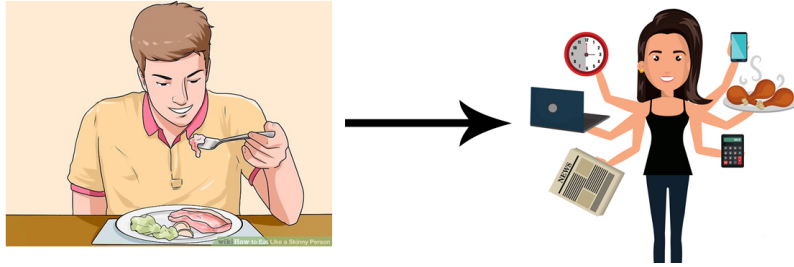


Figure 1.20: Experiments in detecting eating previously considered data recorded in a lab. The field has recently started transitioning to eating in free-living which can be more complicated [79,80].

For reasonable evaluation of a proposed eating activity detection method, it must be evaluated on free-living subjects. This is challenging due to the issue of acquiring ground truth during free-living. To collect ground truth annotations, researchers often accompany participants while they eat in unconstrained or semi-controlled environments or make use of video cameras [71,81]. However, these methods can lead to unnatural behavior on the part of a subject due to the lab coat effect which causes social and surveillance discomfort [82]. When tools used in the lab like video

Table 1.3: Change in precision and recall of meal detection when transitioning from a controlled environment to free-living.

Previous Work	Controlled		Free Living	
	Precision	Recall	Precision	Recall
Thomaz [71]	67	89	65 (-2)	79 (-10)
Mirtchouk [73]	88	87	45 (-43)	85 (-2)
Chun [57]	95	82	78 (-17)	73 (-9)
Zhang [74]	94	90	79 (-15)	77 (-13)

cameras are used in free-living, participants and people they interact are often concerned with their privacy [83]. Subjects attempt to reduce this discomfort by modifying their behavior or by refusing to follow protocol thus leading to unnatural data [82].

Some researchers have created models using data collected in a controlled environment and then tested these models on subjects in free-living, learning that the models suffer a loss in performance (see table 1.3). For example, Merck et al. achieved 88% precision and 87% recall on data collected in a lab [84]. When the experiments were repeated on data collected from subjects in free living, the precision dropped by 43% to 45% [73]. In another experiment, Chun et al. transitioned from data collected in a controlled field study to data collected from free-living subjects, and saw a drop of 17% in the precision, and 9% in the recall of eating events [57]. Similarly, Zhang et al. tested smart eye glasses to monitor chewing and eating. Data was collected from 10 participants who consumed food in the lab, and in free-living. Compared to a precision and recall above 90% in lab studies, their free-living studies obtained precision and recall of 79% and 77%, again showing that behavior in free-living behavior is harder to model and detect accurately [74]. More recently, Doulah et. al collected data from 30 participants in a lab, and 40 participants in an unconstrained environment. Statistical tests showed differences in features such as the number of pauses between ingestion events, the number of bites, the time spent eating, and rate of energy intake, and time spent non-eating [81]. These studies show that eating in free-living is different from eating in a lab, which partly explains why accuracy drops when methods are evaluated on data collected from free-living subjects. Chapter 4 explores this further by studying the effect of secondary activities during eating.

One reason for this drop in accuracy could be that the data for these experiments is collected in controlled or semi-controlled conditions. Classifiers perform better when testing and training data is similar, and thus could perform poorly when training data is collected in a lab, but the method

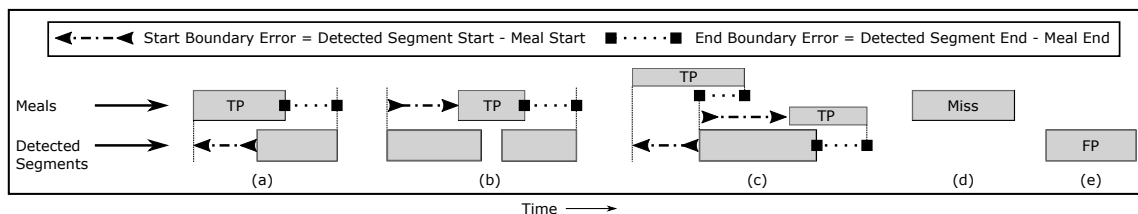


Figure 1.21: Evaluating the performance of detecting eating episodes: (a) one meal overlaps one detected segment, (b) one meal overlaps multiple detected segments, (c) multiple meals overlap one detected segment, and each meal is evaluated for its own TP and boundary error. (d) no detected segment overlaps a meal (miss), and (e) a detected segment does not overlap any meal (FP).

is tested outside it. Data collection in a laboratory is often preferred due to the convenience of obtaining accurate ground truth, however significantly limits the behavior of observed participants. Participants are unlikely to deviate from behavior they have been specifically instructed to perform, thus may not perform activities typically performed while eating in free-living, like talking or watching TV in an unconstrained environment. Ideally, researchers should train and test their models using data collected from free-living subjects. In chapters 2 and 4 we discuss one method of collecting data in free-living, and how it impacts the collection of ground truth.

## 1.10 Metrics for evaluation

A recent review of  $N=40$  works in the field of eating detection tabulated 22 different metrics used in the literature [85]. This shows how nascent the field of ADM is, and how the lack of consistent metrics is a challenge in ADM. We discuss three of the most common subsets of these metrics, and the metrics we report.

### 1.10.0.1 Episode Metrics

Episode metrics describe success at detecting meals. The unit is a meal/snack. These measures quantify success at detecting daily patterns of eating (e.g. three meals at 8 AM, 12 PM and 7 PM) while ignoring total time spent eating. Figure 1.21 shows the 5 different situations that can arise when meals are being detected by classifiers, and how we label detected segments as true positive (TP), false positive (FP), and miss (false negatives). A meal with any amount of overlap with a detected segment is counted as a TP. A meal with no overlap with a detected segment is counted as a miss. A detected segment not overlapping any meal is counted as a FP. True negatives



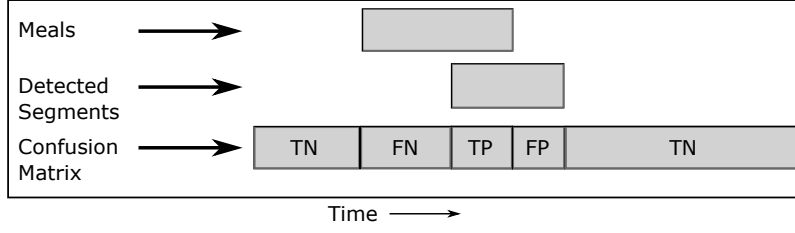


Figure 1.22: Labeling of eating time metrics: true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN).

(TN) are undefined [86]. While it is possible to calculate precision and recall, it is more intuitive to report TPR and FP/TP. These describe the percent of meals successfully detected, and the number of false alarms per true detection. In practice, TPR needs to be high and FP/TP needs to be low for a user to avoid the “cry wolf” effect [87, 88].

#### 1.10.0.2 Boundary Metrics

Boundary metrics describe success at identifying when eating episodes started and ended. The unit is minutes. They are only calculated for meals that are TPs (figure 1.21). We report the average difference between the meal start time and detected segment start time as start boundary error, and the average difference between the meal end time and detected segment end time as end boundary error. In the special case of self-reported meals overlapping multiple detected events (figure 1.21 b and c), for each self-reported meal we use the start/end of the first/last overlapping detection.

#### 1.10.0.3 Time Metrics

Time metrics describe success at labeling every moment of time throughout the day correctly as eating or not eating. The unit is datum. Figure 1.22 shows how we label each datum in the data set as a true positive (TP), true negative (TN), false positive (FP), or false negative (FN). Using these labels we calculate precision, recall (sensitivity), true negative rate (specificity),  $F_1$  score and weighted accuracy ( $ACC_W$ ). However, precision and  $F_1$  score are affected by class balance [89, 90], as they weight classes or recall equally and should only be used on balanced data.  $ACC_W$  offers a more reliable indicator of performance across data sets containing different balances. As non-eating occurs 20 times more than eating during the day, we calculate  $ACC_W$  as shown in the equation below using a weight of 20:1, similar to previous works in ADM [56, 73].

$$ACC_w = \frac{TP \times 20 + TN}{(TP + FN) \times 20 + (TN + FP)}$$

#### 1.10.0.4 Imbalance in Datasets

Eating during free-living occurs infrequently compared to all other non-eating activities. Previous work has often used precision, recall and F1-score to allow comparison between experiments [37, 57, 71], however these metrics are affected by class imbalance. Accuracy can be used as a metric, but since non-eating is appx. 95% of the data, one can blindly label all the data as non-eating, and achieve 95% accuracy. Balanced accuracy is often suggested as an alternative to the previous metrics in experiments with imbalanced classes [90]. In this section we show how precision, F1-score and balanced accuracy are affected by class imbalance, and recommend weighted accuracy instead [56].

The problem with F-score and precision can best be seen by comparing results from tests in which the class distributions change. Consider a classifier that predicts if a given period of time was eating or non-eating. The confusion matrix for this classifier would contain 4 values, as seen in table 1.4. Additionally, these values can be combined to obtain two other values, condition positive ( $P = TP + FN$ ), and condition negative ( $N = TN + FP$ ).

Table 1.4: Example of a confusion matrix for a classifier’s performance.

		Actual Class	
		Eating	Non-Eating
Predicted Class	Eating	True Positives (TP)	False Positives (FP)
	Non-Eating	False Negatives (FN)	True Negatives (TN)

We consider the following metrics:

$$precision = \frac{TP}{TP + FP} \tag{1.8}$$

$$TPR = recall = sensitivity = \frac{TP}{TP + FN} \tag{1.9}$$

$$TNR = specificity = \frac{TN}{TN + FP} \tag{1.10}$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (1.11)$$

$$ACC_B = (TP/P + TN/N)/2 \quad (1.12)$$

$$ACC_W = \frac{TP \cdot W + TN}{P \cdot W + N} \quad (1.13)$$

Where TPR (recall, sensitivity) is true positive rate, TNR (specificity) is true negative rate,  $ACC_B$  is balanced accuracy and  $ACC_W$  is weighted accuracy. They both address the problem of imbalanced data, but  $ACC_W$  allows the imbalance ratio to be fixed to a specific value, in our case 1:20.

To start, consider a dataset containing 100 hours of eating data and 200 hours of non-eating data (1:2 balance). Consider a classifier that has the following results (in hours) on this dataset:

		Actual Class	
		Eating	Non-Eating
Predicted Class	Eating	90	50
	Non-Eating	10	150

The metrics for such a classifier and dataset combination are as follows:

Precision	Recall	TNR	F1	$ACC_B$	Weight	$ACC_W$
0.64	0.9	0.75	0.75	0.825	2	0.825

Now consider the same classifier operated on a dataset with twice as much data in the non-eating class (400 hours, 1:4 balance). We assume the classifier would perform similarly on the added data, leading to doubling of the number of true negatives and false positives.

		Actual Class	
		Eating	Non-Eating
Predicted Class	Eating	90	100
	Non-Eating	10	300

This results in the metrics changing to:

Precision	Recall	TNR	F1	$ACC_B$	Weight	$ACC_W$
0.47	0.9	0.75	0.62	0.825	4	0.825

In a dataset with a 1:20 balance, the drop in precision and F1 score is even larger:

Precision	Recall	TNR	F1	$ACC_B$	Weight	$ACC_W$
0.15	0.9	0.75	0.26	0.825	20	0.825

Recall (TPR), TNR, balanced accuracy and weighted accuracy are the same for all three examples, regardless of imbalance. However, balanced accuracy, precision, and F-score are undefined when there are no examples in the positive class i.e.  $P = 0$ . This can happen in free-living on a day where a subject does not consume any food. To address this issue, previous works have sometimes discarded participant data when the participant did not consume food.

When such outliers are not discarded, other work has used weighted accuracy which does not suffer from these issues [47, 56]. Following suit, for this work we will report the time metrics of TPR (Recall), TNR and weighted accuracy, since the high amount of skew in free-living datasets results in questionable performances for other metrics.

## 1.11 Contributions of this work

First, we collected a data set an order of magnitude larger (351 subjects) than all the previous works cited above (4-43 subjects). We call this data set the Clemson All-day data set (CAD). All the data was collected during free living, with a 20:1 ratio between non-eating and eating. We show that changes in the sensor platform used for data collection affect not only the pre-processing of the collected data, but also the performance of automatic methods to detect eating. While analyzing data from a newer, less restrictive sensor package, we learn that there is no standard for IMU data. Some devices may report only report raw acceleration which is different from linear acceleration reported by the iPhone used by Dong et al. [56]. There are also no standard coordinate systems, thus different devices report the sensor data differently.

We show that while linear acceleration of the wrist can be estimated, noise and bias errors lead to values that are not feasible in wrist motion. For the first time, we quantify the amount

of noise in linear acceleration estimates of the wrist. We show that this noise range overlaps the range of true wrist acceleration, explain the sources of this noise, and demonstrate a method to mitigate the noise. We show how this noise affects the peak detection and eating activity detection algorithms.

We analyze information on secondary activities conducted during a meal, quantify how often these activities occur in free-living. We show that secondary activities are often multitasking activities. This multitasking leads to overlapping and intermittent secondary activities, causing difficulty modeling such meals. We develop a classifier to detect two common secondary activities: walking and resting. These two activities exhibit motion patterns that are distinctive enough from eating that they can be reliably detected as sub-activities within periods of time that subjects report as meals or snacks. By augmenting an eating detection classifier to include walking and resting, we were able to improve weighted accuracy from 74% to 77% on our free-living data set ( $t[353]=7.86$ ,  $p<0.001$ ), showing evidence of how secondary activities can impact the detection of eating. These results help explain why accuracy drops when methods to detect eating are tested in free-living, and how detecting secondary activities is key to improved detection of periods of eating.

Finally, we use this knowledge of eating activities to create a CNN based classifier to detect periods of eating. Previous work in the literature have taken a bottom-up approach to detecting eating episodes, starting with the detection of individual ingestion events such as bites, swallows or chews using a 1-5 second window [45, 86, 91–93]. This is sometimes called gesture spotting [91, 94]. Detected events are then aggregated to identify periods of eating (meals) [86]. We take a top-down approach by analyzing a much longer window of 0.5-15 min. During a meal, we expect the window to contain gestures that are related to eating but that are not necessarily ingestion events. Examples include preparing foods for consumption, stirring, cutting, and resting between ingestion events [54, 95]. We evaluate this idea across a range of window sizes and report accuracy on the Clemson all-day dataset (CAD) which is the largest publicly available data set of its kind.

## Chapter 2

# An experiment to replicate previous work on meal detection

As summarized in chapter 1, Dong et al. demonstrated a method that identified periods of eating with a weighted accuracy of 81% [56], at a weighted sensitivity of 81% and a weighted specificity of 82%. A total 449 hours of data was identified for use from 43 subjects. This data contained 22.4 hours of eating in 116 periods of eating. While commercial activity trackers were available in a small form factor suitable for the wrist, gyroscopes, an important sensor for the above algorithm, were not a part of these trackers due to the high current consumption by gyroscopes at that time. Therefore, wrist tracking data was collected using an Apple iPhone 4 custom programmed to log acceleration and angular velocity. An arm band was used to strap the phone to the wrist, as shown in figure 2.1. It is questionable how this device impacted the collected data for this experiment. The iPhone4 weighs 137g, similar to the weight of a dress watch. This weight and device shape, along with the cumbersome armband might have restricted movement and limited the range of wrist gestures executed in free-living. Since mounting mobile phones on the wrist is uncommon, it is unknown how it physically burdened subjects at the time of data collection, and if this burden affected the usual spread of activities performed in a day. For example, a subject might not move the wrist as much during eating due to the device weight. They may also decide not to play tennis being concerned that the armband strap might fail and the iPhone might fall.

It is also questionable if data from 43 subjects is enough to cover the variability in wrist



Figure 2.1: The iPhone4 mounted on the right arm for data collection [56].



Figure 2.2: Photo comparing size of Shimmer3 with iPhone mounted on the wrist.

motion exhibited in the general free-living population. Several previous works have seen drops in performance in the range of 2-11% (recall) when transitioning from controlled testing to free living [57, 71, 73, 74]. A larger data set could help answer some of these questions.

In this chapter we discuss the replication of Dong’s method to detect eating by tracking wrist motion [56]. We differ from the original experiment in a few ways. First, we use newer sensors available in a smaller, convenient wrist watch like package, the Shimmer3 (section 2.1) to collect, analyze and evaluate a new data set, the Clemson All-day data set (CAD). A major difference between the two devices is the weight. The Shimmer3 weighs 24g, and is much lighter than the iPhone which weighs 137g. Figure 2.2 compares the sizes of the Shimmer3 and the iPhone 4 when mounted on the wrist. This  $\approx 5x$  reduction in weight of the device might enable more varied movements at the wrist, which may result in different sensor data during eating. Using the Shimmer3 also led to changes in the way data was collected and processed. The Shimmer3 and the iPhone do not report acceleration in the same format. While the iPhone provided linear acceleration of the wrist which is used by the eating detection algorithm, the Shimmer3 does not provide it, instead providing raw acceleration, a combination of the gravity field and the linear acceleration experienced by the device. Section 2.2 explains how linear acceleration was obtained from Shimmer3’s raw acceleration. Another difference between the iPhone and the Shimmer3 is the use of different coordinate systems. Since there is no unified convention for the coordinate systems, we define a unified wrist coordinate system. We then transform data collected by the Shimmer3 and the iPhone to the wrist coordinate system before processing. This step is described in section 2.3.

A second difference compared to the original experiment by Dong et. al, is the size of the dataset. 408 subjects were recruited to collect data for CAD, an order of magnitude larger than [56]. We describe the issues faced and the lessons learned from this data collection. Lastly, we compare the accuracy of this replication experiment against the original experiment to determine the effects of the larger sample size and the change in device form factor.

## 2.1 Shimmer3 Sensor Module

The eating activity detection algorithm assumes that a person is wearing wrist mounted accelerometer and gyroscope sensors that provide linear acceleration  $(l_x, l_y, l_z)$  and angular velocity  $(\omega_\phi, \omega_\theta, \omega_\psi)$ . Previous work attempted to build a data logger in-house [96]. The purpose of such





Figure 2.3: Photo of the Shimmer3 IMU unit [62].

a device was to replace the iPhone as the data collection platform in a new large year-long data collection. We learned that this long term data collection would require 20 - 25 such devices. Due to the requirement of this quantity of devices, and the time frame for procurement, we considered smartwatches and commercial activity trackers such as the Fitbit, Apple iWatch and the Samsung Gear. However, at the time of experiment design these devices either lacked a gyroscope, did not expose raw sensor data, or have the memory to store data for an entire day. After survey, an early version of the Shimmer3 [62] from Shimmer Sensing was selected for data collection.

The Shimmer3 houses a PCB mounted with a MSP430 microcontroller that can be programmed to log data from sensors onto a microSD card. Two multicolor LEDs and one button are available for user interaction, such as setting markers or turning the recording on or off. The on-board MPU-9250 MARG sensor provides IMU and orientation information processed by a proprietary algorithm. Survey revealed that the device contained all the required sensors, and could record data continuously for multiple days, and Shimmer Sensing offered hardware and software support if needed.

The device can be configured using the Consensys software provided by Shimmer Sensing, or by using a configuration text file for advanced control. Figure 2.4 shows a screenshot of Consensys showing the configuration screen. Through this screen a user can select the sensors they would like

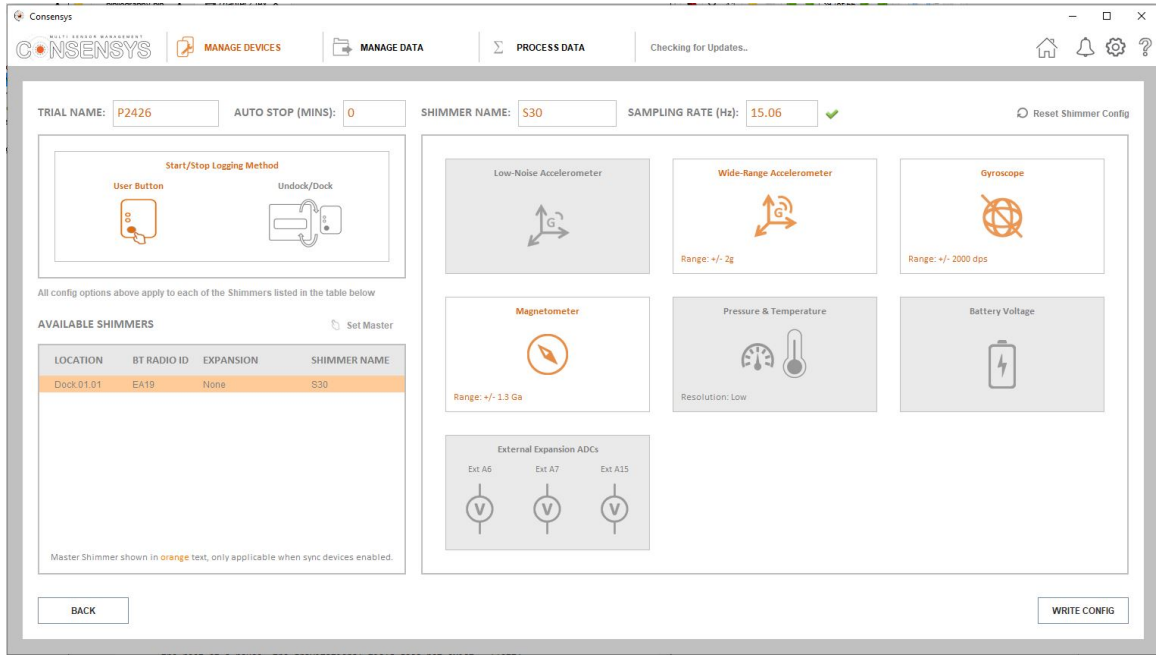


Figure 2.4: Screenshot of the configuration screen in Consensys.

to log. This screenshot also shows the sampling rate set to 15.06 Hz, with a check mark next to the number. This is because the Shimmer3 firmware can not sample at exactly 15 Hz, and suggests a sampling rate of 15.06 Hz. Data recorded by the Shimmer3 is read by the Consensys software, which then exports a CSV file containing sensor data. The CSV file contains data from the sensors, and time stamps with millisecond precision. CSV data at 15.06 Hz was resampled to 15 Hz before processing further.

## 2.2 Linear Acceleration of the wrist

The algorithm presented in this work processes linear acceleration, which is different from raw accelerometer data. While the latest version of the Shimmer3 firmware can provide linear acceleration estimates through an offline algorithm, the Shimmer3 algorithm available at the time of data collection only provided raw acceleration values (a combination of gravity and linear acceleration), leaving the estimation of linear acceleration to the user. Estimating linear acceleration is complicated due to the presence of sensor noise. We first discuss why sensors cannot sense linear acceleration, and then how linear acceleration can be estimated.

While studying gravity, Einstein noted that an observer watching a ball fall under gravity

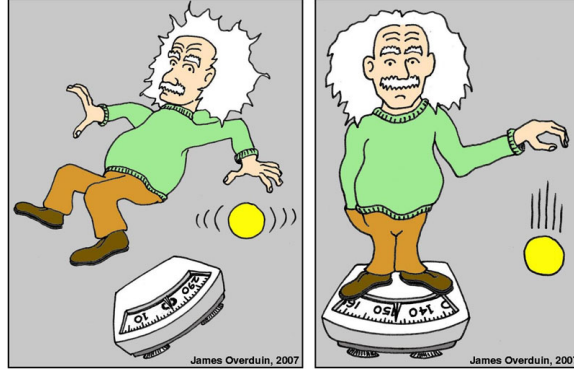


Figure 2.5: Einstein’s happiest thought (1907): “For an observer falling freely from the roof of a house, the gravitational field does not exist” (left). Conversely (right), an observer in a closed box—such as an elevator or spaceship cannot tell whether his weight is due to gravity or acceleration. [97]

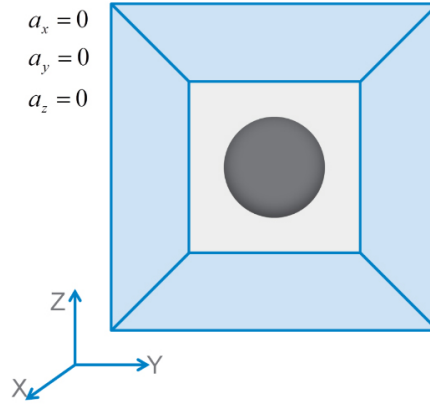


Figure 2.6: An accelerometer in free-fall [62].

can transform gravity to acceleration by moving to the ball’s frame of reference. A consequence of this phenomena is that inside a box, an observer cannot tell if the acceleration they are experiencing is due to a gravitational field, or due to an external force acting on the box.

Similarly, accelerometers are unable to tell the difference between the effect of gravity and acceleration caused due to the movement of the wrist. Thus, an accelerometer reading is the result of the superposition of the gravity vector and linear acceleration of the device. Acceleration sensed by MEMS accelerometers is commonly explained by using examples of a ball shaped sensor of mass  $M$  inside a cube. The Shimmer user guide [62] shows an example of the ball inside a cube during free-fall (figure 2.6). During free-fall, the cube does not exert any force on the ball inside it, and thus each axis of the sensor will report an acceleration of  $0g$ .

In a state where the cube is lying at rest on a surface, the ball will rest against the lower

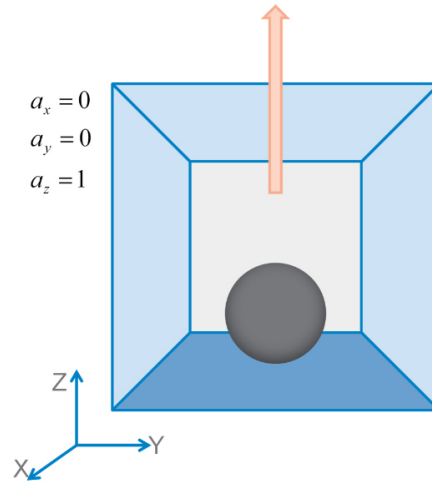


Figure 2.7: An accelerometer at rest on a surface, measuring gravity [62].

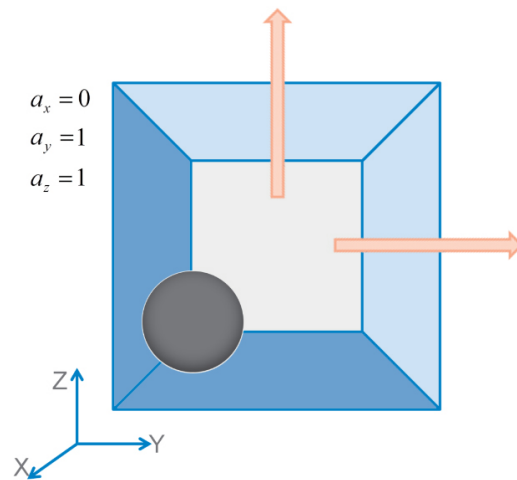


Figure 2.8: An Accelerometer resting on a surface being pushed to the right. (derivative from [62])

wall of the cube due to gravity. The walls of the cube will exert a force equal to that of the weight of the ball, pushing against it upwards. Thus the ball experiences a force of  $M \times 1g$ , resulting in the accelerometer reporting an acceleration of  $+1g$ . If this cube at rest was to be pushed upon, or accelerated to the right with an external acceleration of  $1g$ , the sensor would then report an acceleration of  $A_y = 1g$  on the Y axis, and  $1g$  on the Z axis. For another example, consider rotating the cube by  $45^\circ$  and letting it rest on a corner. This would result in reactive force from two walls, with the sensors reporting an acceleration of  $1g \cos 45^\circ = 0.707g$  on the Y and Z axis.

Due to this behavior, linear acceleration must be estimated from accelerometer readings. This can be done by first estimating the orientation of the device as rotation matrix  $R$ . Estimating orientation from IMU data is ongoing research, and various filters exist to achieve this [98–100]. For this work, we rely on orientation estimates provided by the InvenSense MPU-9150 housed in the Shimmer3. If the estimation of the device orientation is represented by the rotation matrix  $R$ , gravity  $G$  in the device frame can be calculated from  $R$  and gravity in earth frame  $G_e$  as:

$$G = [R \cdot G_e] \quad (2.1)$$

Finally, in the device frame, linear acceleration  $L$  can be estimated by removing gravity  $G$  from the raw accelerometer values  $A$ :

$$l = A - G \quad (2.2)$$

The InvenSense MPU-9150 onboard the Shimmer3 provides orientation estimates in the form of quaternions, which must be converted to a rotation matrix. We briefly discuss some representations of rotation and their benefits or disadvantages.

### 2.2.1 Representing orientation

Rotations or orientations along a single axis can be represented by specifying an axis, and the amount of rotation around that axis. This is known as the Euler axis and angle notation. The angle can be represented as a three dimensional unit vector, and the angle by a scalar. However, when the scalar is 0, the axis is not uniquely defined. Additionally, combining two successive rotations is not straightforward as the angle axis notation does not follow the law of vector addition, as rotations are not vectors. Euler angles are used by aircraft systems as they are simple and intuitive to understand.

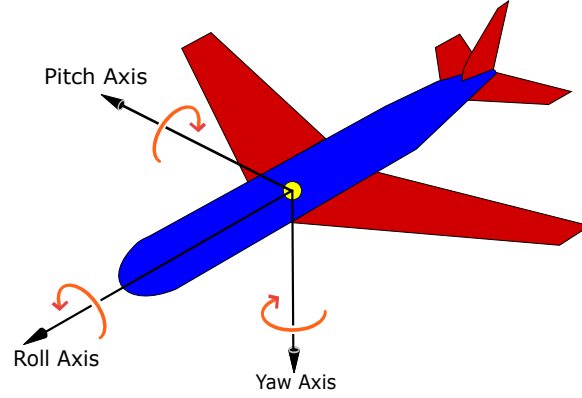


Figure 2.9: Yaw, pitch and roll Euler angles for an airplane [101].

Fig 2.9 shows an example of an airplane and its three angular axis for explanation. However, Euler angles and the angle axis notation suffer from gimbal lock, a phenomena where orientation cannot be measure or computed when the pitch angle approaches  $\pm 90^\circ$ , as  $\tan(90^\circ)$  is undefined.

An alternative is the rotation matrix, where 9 parameters describe the transformation of a vector from one rotational configuration to another. The ease of using matrices, and the ability to combine successive rotations make the rotation matrix a popular way to represent rotations in the literature. The following matrices show example of rotation by  $\theta$  about the three axis.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Quaternions are another alternative form to represent orientation or rotations. Quaternions are an extension of the complex number system. While complex numbers can be represented in the form of  $a + b \mathbf{i}$  where  $a$  and  $b$  are real numbers, the general form of quaternion representation is  $a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} = [b_i \ c_j \ d_k \ a_r] = [q_i \ q_j \ q_k \ q_r]$ , where  $a, b, c, d$  and  $q_i, q_j, q_k, q_r$  are real numbers while  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  are the fundamental quaternion units. When used to represent rotation, if the Euler angle representation of a rotation was given by the angle  $\theta$  and the Euler axis  $\hat{\mathbf{e}} = [e_x \ e_y \ e_z]$ , the quaternion representation would be:

$$q_i = e_x \sin \frac{\theta}{2} \quad (2.4)$$

$$q_j = e_y \sin \frac{\theta}{2} \quad (2.5)$$

$$q_k = e_z \sin \frac{\theta}{2} \quad (2.6)$$

$$q_r = \cos \frac{\theta}{2} \quad (2.7)$$

Successive rotations can be combined using quaternion multiplication, for example  $q_3 = q_2 \otimes q_1$ . Quaternions are the preferred notation when computing is involved as quaternions are more compact than matrices, conversion from rotation matrices to quaternions and vice-versa involves no trigonometry, and more importantly, computational costs for quaternions are much lower than those for a 3 x 3 matrix.

Due to the reasons of computational efficiency and compactness, the Invensense MPU-9150 provides orientation data in the form of quaternions. The quaternion representation can be converted into a rotation matrix  $R$  using the following equation:

$$R = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \quad (2.8)$$

Where  $q_0, q_1, q_2$  and  $q_3$  are equivalent to  $q_r, q_i, q_j$  and  $q_k$ . Gravity can be obtained from this matrix representation using a single equation, 2.9 shown below:

$$G = [R \cdot G_e] \quad (2.9)$$

## 2.3 Rotating Coordinate Axis

While the above orientation representation allows us to obtain gravity and thus linear acceleration, the coordinate axes on the Shimmer3 (x,y,z, yaw, pitch, roll) do not match those of the iPhone. Thus features depending on a specific axis such as wrist roll will not be calculated correctly. We must therefore rotate axis correctly before processing this data. Wrist mounted devices such as the iPhone, Shimmer3, Actigraph, Apple iWatch and the Samsung Gear define their coordinate

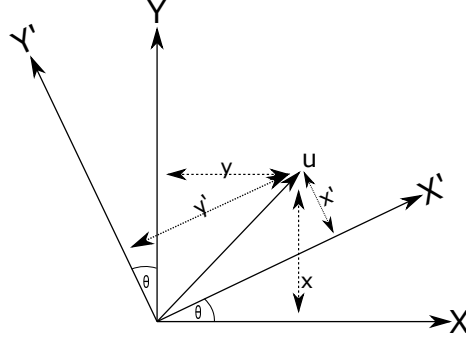


Figure 2.10: Two coordinate axes: O and O', with O' rotated by  $\theta$  relative to O.

systems independently, not following any unified convention. The algorithm by Dong et al. used coordinate axes defined for Apple iPhone, and thus data collected by the Shimmer3 must be rotated to this coordinate system. For the purpose of this experiment and future research by the research group, we define a wrist coordinate system, and provide an example of how accelerometer and other sensor data can be rotated from an arbitrary coordinate system to the wrist coordinate system.

The following explanation on transformations between two 2D coordinate systems is adapted from [102]. We define O using axes X and Y, and O' using X' and Y'. The two systems share the same origin, O. O' can be obtained from O by a rotation of  $\theta$ . A vector u can be represented in O as  $(x, y)$  and in O' as  $(x', y')$ . Figure 2.10 shows this setup graphically. The relation between  $(x, y)$  and  $(x', y')$  is as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (2.10)$$

To obtain the representation of a vector in O' given the representation in O, we pre-multiply equation 2.10 by the inverse of Q,  $Q^{-1}$ . This results in equation 2.11:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.11)$$

Similar to 2D rotations, 3D rotations along a single axis can be transformed using a rotation matrix,



with the rotational matrices defined for the three axis as follows:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

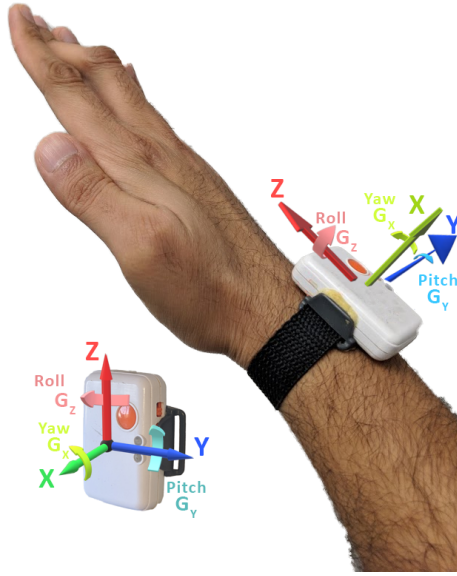


Figure 2.11: Accelerometer and gyroscope coordinate axes for the Wrist coordinate system

Multiple rotations can be achieved by multiplying the rotation matrices in the correct order. For our research, we have consider two coordinate systems. The wrist coordinate system (fig 2.11) is defined by us, while the Shimmer3 coordinate system is defined in the Shimmer user guide (fig 2.13a). Previous work by Dong et al. demonstrated that rotations in wrist roll were the strongest indicators of bites and periods of eating, thus the literature named the three axes as pitch, roll, yaw. However, sensor manufacturers define linear and rotational axes using the X,Y,Z notation. To unify these naming conventions, we combined the X,Y,Z and yaw, pitch, roll notations. We first assigned the yaw, pitch and roll axes by comparing wrist movement to that of an aircraft (fig 2.11). The X,Y,Z axes are then set to correspond to yaw, pitch and roll. The right hand rule (fig 2.12) is followed to set the positive directions for these axes.

The data recorded by the Shimmer3 must be transformed to the wrist coordinate system.

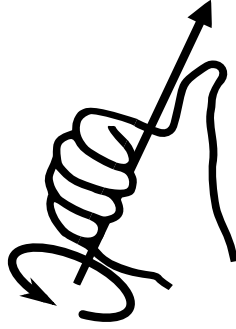


Figure 2.12: The right hand rule state that when the fingers curl in the direction of rotation, the thumb points towards the positive end along the axis of rotation [103].

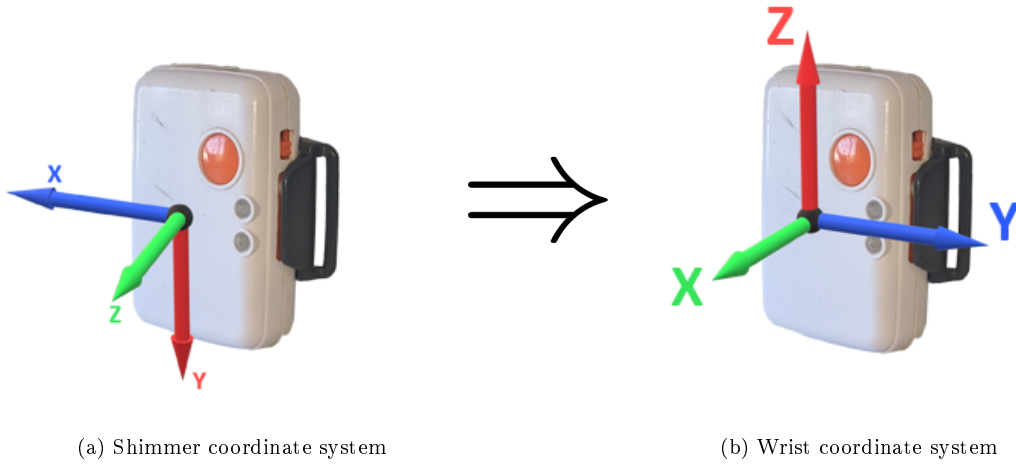


Figure 2.13: Data from the Shimmer has to be transformed to the wrist coordinate system.

This rotation can be achieved by first rotating by  $-90^\circ$  along the Y axis using matrix  $R_1$ , followed by a rotation of  $90^\circ$  along the new X axis using matrix  $R_2$ , where  $R_1$  and  $R_2$  are shown in equations 2.13 and 2.13.

$$R_1 = R_y(-90^\circ) = \begin{bmatrix} \cos -90^\circ & 0 & \sin -90^\circ \\ 0 & 1 & 0 \\ -\sin -90^\circ & 0 & \cos -90^\circ \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (2.13)$$

$$R_2 = R_x(90^\circ) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 90^\circ & -\sin 90^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2.14)$$

Rotations  $R_1$  and  $R_2$  can be combined into a single vector rotation  $R_c$ :

$$R_c = R_1 * R_2 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \quad (2.15)$$

To convert a vector from the Shimmer3 CSV coordinate system to the wrist coordinate system, we multiply the vector by  $R_c^{-1}$ :

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R_c^{-1} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.16)$$

equation 2.16 can be simplified:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} Z \\ -X \\ -Y \end{bmatrix} \quad (2.17)$$

Thus, equation 2.17 can be used to rotate sensor data from the Shimmer3 system to the wrist coordinate system, and similar transformations can be used for any other device.

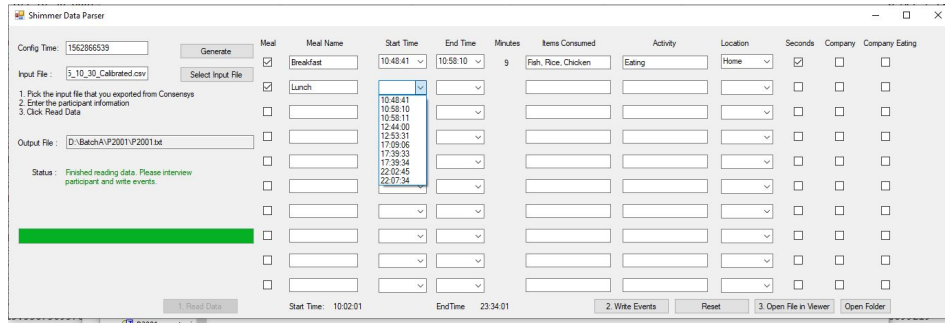


Figure 2.14: Screenshot of MarkerParser, the tool used to process Shimmer CSV files and collect interview information from participants.

## 2.4 Software for Data Collection

The off the shelf Shimmer3 device and accompanying software required modifications before it would be suitable for data collection. As discussed above, pressing the button on the Shimmer would result in starting or stopping of data collection, and there was no support for recording time stamps or meal markers. We custom programmed the device to add the ability of recording meal start and stop times using the button available on the Shimmer3. The device was reprogrammed to start and stop recording data when the button was pressed longer than 3 seconds. To record a marker to indicate the start or stop of a meal, the button could be tapped. To store the information of this timestamp, we overloaded the data stream of the galvanic skin response (GSR) sensor in the Shimmer3 firmware. Since the GSR sensor was absent in our data collection, the value of this sensor was always reported as  $-40$  kOhms. We programmed the firmware to change this value to  $100$  kOhms when the button was tapped. By matching the timestamp to the GSR value of  $100$  kOhms in the CSV file created by Consensys, button tap information could be procured.

Additionally, the Shimmer3 did not support sampling at  $15$  Hz, and thus the data needed to be resampled. The data collected by the device did not contain linear acceleration information, and thus we had to estimate linear acceleration, and lastly we had to rotate the coordinate axes of the collected data to match the wrist coordinate system. These processes, as well as button tap information was processed by a custom software called MarkerParser, shown in figure 2.14. Additionally, MarkerParser was used to collect information contextual information on the meals, and confirm self reported meal start and end times. Figure 2.14 shows a drop down menu populated from button tap information from the Shimmer device. A subject was interviewed with this information to confirm meal boundaries.

## 2.5 Data Collection

We collected a data set we call the Clemson All-day data set (CAD). Data was collected over a period of one year. The Clemson University Institutional Review Board approved data collection and participants were provided informed consent. Participants were recruited from the student, faculty and staff bodies of Clemson University, and the general population near the city of Clemson. Data was collected through a three day protocol, where subjects were trained on day one, recorded data on day two, and returned the device and data on day three. On day one, participants were provided with an interview, collecting the details on their eating habits, along with their gender ethnicity, weight, height, age and BMI. The participants were trained on operating the device through a 3 minute demo interaction and provided instructions on self-reporting meals.

Each participant was instructed to wear the device on waking up on day 2, and long press the button to start recording data. The subjects were requested to tap the button before the start of any period where the primary activity was eating, such as a meal or a snack, or any other contiguous period of eating. Subjects were instructed not to timestamp any grazing consumption activity such as drinking coffee while working or reading a book over a long period of time, where the primary activity may not be eating. At the end of the day, participants were instructed to stop recording data, and take off the device. On day 3, participants returned to the lab for an exit interview. A researcher downloaded the data from the device and interviewed the participant using time stamps from button presses. Erroneous button presses were ignored, and time stamps were paired to identify meal start and end times. Secondary information on the periods of eating was also collected such as the location (eg. home, restaurant or other), type (breakfast, lunch, dinner or snack), if the eating activity was consumed in a single sitting or a second serving was acquired (yes or no), if the subject consumed food in the company of another person (yes or no), and if the person was also consuming food (yes or no). Open response questions were asked to collect information on what food was consumed, and if any activity was being performed during this period. After collecting this data, the researcher loaded it into a custom C program called Phoneview, and confirmed visually there were no issues such as missing data, or a device crash.

A total of 408 subjects were recruited (61% female, BMI  $25.8 \pm 5.8 \frac{kg}{m^2}$ , age  $28 \pm 12$  years), and data was recorded on all seven days of the week. Table 2.1 shows participant demographics in detail. Of the 408 subjects recruited, 351 (86%) completed a recording that was usable. Recordings were

Table 2.1: Demographic details for participants recruited for CAD

<b>n</b>	
351	
<b>Age</b>	
mean ( $\pm$ SD)	28 $\pm$ 12 years
<b>Gender</b>	
Female	214 (61%)
Male	137 (39%)
<b>Ethnicity</b>	
Black	69 (20%)
White	205 (58%)
Other	77 (22%)
<b>BMI</b>	
mean ( $\pm$ SD)	25.7 $\pm$ 5.73 $\frac{kg}{m^2}$

Table 2.2: Number of meals in the dataset by day of week.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
84	132	390	284	161	51	30

unusable for the following reasons: 19 device failures (software crash, firmware issues), 4 subjects forgot to turn device on, or inadvertently turned the device off, 2 subjects carried the device in their pocket instead of wearing it on their wrist, 2 devices were lost or damaged, and 13 subjects failed to follow instructions. All these events led to the loss of data. One subject recorded for three days, and another subject recorded for two days, leading to a total of 354 days of usable data collected. A total of 4,680 hours of data was collected, containing 265 hours of eating activity across 1,133 separate eating activities. Table 2.2 lists daywise counts of eating activity records in the dataset. The average amount of data recorded per participant was 13.2 hours. The average start time for a recording was 8:50 am, while the average end time for recordings was 22:06 pm.

## 2.6 Detection of Eating from Wrist Motion Data

The method assumes that a person is wearing wrist mounted accelerometer and gyroscope sensors that provide linear acceleration ( $l_x$ ,  $l_y$ ,  $l_z$ ) and angular velocity ( $\omega_\phi$ ,  $\omega_\theta$ ,  $\omega_\psi$ ). [56] learned that wrist activity peaks before and after the consumption of a meal, and hypothesized that this is caused due to actions conducted before a meal like walking, procuring food and bringing it to a

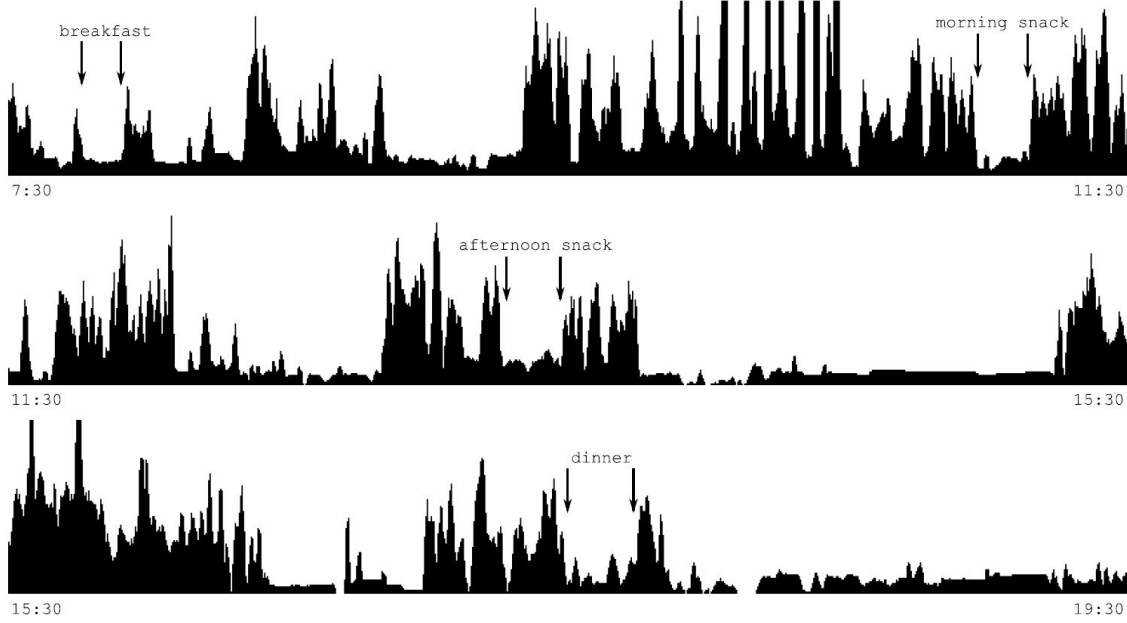


Figure 2.15: Wrist motion activity from one day of recording in CAD. Meals logged are indicated with arrows marking start and end times.

table, or clean up after meal consumption. Examples of this can be seen in figure 2.15, which shows 12 hours of wrist motion activity from a single participant. Note how wrist activity is low for the breakfast, snacks and dinner, while it shows peaks before and after these activities. The figure also includes the start and stop times of meals indicated as arrows. Peaks in wrist motion energy are used to segment periods of time. For each segment, four features are calculated that are then used to classify the segment as eating or non-eating using a Naive Bayesian classifier.

### 2.6.1 Pre-processing

Each signal from the accelerometer ( $l_x, l_y, l_z$ ) and gyroscope ( $\omega_\phi, \omega_\theta, \omega_\psi$ ) is filtered independently using a centered Gaussian-weighted window to reduce the effect of noise as shown below:

$$S_t = \sum_{i=-N}^0 R_{t+i} \frac{\exp(\frac{-t^2}{2\sigma^2})}{\sum_{x=0}^N \exp(\frac{-(x-N)^2}{2\sigma^2})} \quad (2.18)$$

where  $R_t$  represents raw signals  $\{l_x, l_y, l_z, \omega_\phi, \omega_\theta, \omega_\psi\}$ , and  $S_t$  represents smoothed signals  $\{S_{x,t}, S_{y,t}, S_{z,t}, S_{\phi,t}, S_{\theta,t}, S_{\psi,t}\}$  at time  $t$ . The window is of length  $N=1$  seconds and operates on past data with a Gaussian  $\sigma$  of 10s.

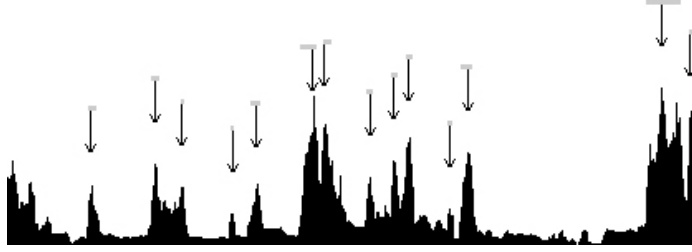


Figure 2.16: First two hours of data from figure 2.15, showing peaks detected by the detection algorithm with arrows.

### 2.6.2 Segmentation

Peaks in wrist motion energy are used to segment the data over time. This energy ( $E$ ) can be characterized by the sum of acceleration, calculated as:

$$E_t = \frac{1}{W+1} \sum_{i=t-\frac{W}{2}}^{t+\frac{W}{2}} |S_{x,t}| + |S_{y,t}| + |S_{z,t}| \quad (2.19)$$

where  $W$  is the window size of 1 minute, shown to be suitable for detecting peaks in wrist motion activity caused by long motions indicative of eating boundaries, while smoothing over shorter motions. Peaks can be observed at the boundary of these peaks, with reduced wrist motion energy during the meals themselves. These peaks are detected by using a hysteresis based peak detector [56]. Figure 2.16 from [56] shows an example the detected peaks. Each peak-to-peak period is then set as a segment for classification.

### 2.6.3 Features

Four features identified to be useful in classifying segments as eating and non-eating are calculated for each peak-to-peak segment. The first feature is called manipulation:

$$f_{1,w} = \frac{1}{W} \sum \frac{|S_{\phi,t}| + |S_{\theta,t}| + |S_{\psi,t}|}{|S_{x,t}| + |S_{y,t}| + |S_{z,t}|} \quad (2.20)$$

where  $W$  is the length of the segment, and  $S$  is the smoothed datum of the respective signal. This feature measures the ratio of rotational motion to linear motion, which is higher during periods of eating due to frequent rolling of the wrist.

The second feature is the sum of acceleration, which characterizes the amount of linear



motion in the wrist. Similar to wrist motion energy, this feature has lower values during periods of eating, and higher values during periods of vigorous activity.

$$f_{2,w} = \frac{1}{W} \sum^W |S_{x,t}| + |S_{y,t}| + |S_{z,t}| \quad (2.21)$$

The third feature wrist roll is calculated as the deviation from the mean of wrist roll over the segment:

$$f_{3,w} = \frac{1}{W} \sum^W |S_{\phi,t} - \frac{1}{W} \sum^W S_{\phi,t}| \quad (2.22)$$

The wrist exhibits frequent rolling during periods of eating as food is repeatedly brought to the mouth. This is captured by the fourth feature which calculates wrist roll motion as:

$$f_{4,w} = \frac{1}{W} \int_W 1_{\forall t \in [|S_{\phi,t}| > 10^\circ \dots t + 8\text{sec}]} \quad (2.23)$$

This feature ranges between 0 and 1, representing the percentage of time the wrist is in roll motion. It is calculated by including the period of time the wrist roll velocity is at least  $10^\circ/\text{sec}$ , plus 8 seconds after the velocity drops below  $10^\circ/\text{sec}$ . These values were found optimal in previous work on detecting bites during a meal [56].

#### 2.6.4 Classification

Each segment is classified as eating or non-eating using the Naïve Bayes classifier, which assigns the most probable class given feature values  $f_j$ . Assuming independence of features, the class can be assigned as:

$$c_i = \arg \max_c P(c_i) \prod_j P(f_j | c_i) \quad (2.24)$$

For the problem of detecting eating, we have only two classes: eating ( $c_0$ ) and non-eating ( $c_1$ ). We tested prior probability values  $P(c_0), P(c_1)$  between  $\{0.05, 0.95\}$ ,  $\{0.10, 0.90\}$ , ...  $\{0.95, 0.05\}$ , and found the best performance at  $P(c_0) = P(c_1) = 0.5$ . For each feature, the probabilities

were modeled using a normal distribution:

$$P(f_j|c_i) = \frac{1}{\sqrt{2\pi\sigma_{i,j}^2}} \exp\left(-\frac{(f_j - \mu_{i,j})^2}{2\sigma_{i,j}^2}\right) \quad (2.25)$$

where  $\mu_{i,j}$  is the mean of feature  $j$  in class  $i$  and  $\sigma_{i,j}^2$  is the variance.

### 2.6.5 Training

We used leave one out cross validation for testing, i.e. for testing each of the 354 recordings in CAD, the classifier was trained using the other 353 recordings. The training process finds the parameters  $\mu_{i,j}$  and  $\sigma_{i,j}^2$  for each class labeled by the Bayesian classifier. For the eating class, we considered all ground truth meals. The value of each feature was calculated from self-reported start time to self-reported end time for each meal. The mean and variance was then calculated over all meals. For non eating, the data was split into 5 minute chunks. Feature values were calculated for each chunk, and then the mean and variance was calculated over all 5 minute chunks.

### 2.6.6 Evaluation metrics for the segmenter

The peak detection based segmenter acts as a bottleneck when evaluating the eating detection algorithm. Put simply, if the segmenter accuracy is X%, the accuracy of the classifier cannot be  $\geq X\%$ , as the classifier labels periods segmented by the peak detector.

The peak based segmentation method assumes that peaks of wrist motion activity happen before and after a meal. Dong’s experiment showed that this was common in the data collected for their experiment [56]. Across 43 subjects, the average time between self-reported meal start times and correctly detected EA was -0.6 minutes. The average time between self-reported meal end times and correctly detected EA was +1.5 minutes [56].

Using CAD, we evaluate the performance of the peak based segmenter similarly, by evaluating the time difference between self-reported meal start and end times, and the nearest peaks to them.

### 2.6.7 Evaluation metrics for the classifier

We evaluate the classifier in two ways, which can be described by two questions. The first question we ask is “How many eating activities were correctly identified by the classifier?”. Detecting eating activities assists researchers working with 24 hour recalls, as subjects can be reminded of when they were eating, and interviewed, leading to more accurate data. Alternatively, when an eating activity is detected, researchers can trigger interventions or ecological momentary assessments (EMAs), which help collect accurate data. To answer this question, we use two metrics, the % meals detected, and the % duration detected.

The second question we ask is “What was the accuracy of the classifier, when considering duration?”. Duration is an important assessment as it is related to the eating habits of a subject, and also correlated to the amount of food consumed in a meal. For this we can use weighted accuracy (WACC), which is defined as:

$$WACC = \frac{TP \times 20 + TN}{P \times 20 + N} \quad (2.26)$$

where WACC is the weighted accuracy, TP is the number of seconds that were correctly identified as eating, TN are the number of seconds correctly labeled as non-eating, P are the total number of seconds reported as eating, and N are the total number of seconds reported as non-eating. WACC relies on the accuracy of self reported start and stop times of meals, and may include periods when the subject was not eating. A 20:1 factor was used in the calculation of WACC due to the class imbalance in eating and non-eating. Annual labor surveys have reported that Americans spent an average of 1.17 hours daily eating and drinking,  $\approx$  1:19 [70]. This was reflected in work by Dong et al. where 22.4 hours of 449 total hours were eating activities, and CAD, where the ratio is appx. 1:20.

## 2.7 Results

From the data collection, we learn that 14% of the recordings did not provide a recording that was usable. One subject lost the device at the neighborhood recreation center, while another subject’s dog chewed on the device, leading to no usable data from the two subjects. 2 subjects did not wear the device on their wrist, but instead carried it in their pockets. 4 turned the device off inadvertently, and 13 subjects failed to follow instructions, for example only recording data during

eating, thus not collecting any data for the non-eating class. This shows that a wearable device like a wrist watch can be expected to fail 9.3% of the time due to human behavior and unpredictable events. The device also failed 19 times due to software crash, or incorrect setup parameters alluding to an expected fail rate of 4.6% due to hardware issues.

### 2.7.1 Change in Feature Values

Table 1 lists the means, variances and fishers linear discriminant (FLD) values for the four features in the two classes and datasets. Changes in separation across datasets can be compared by comparing the separation between the means and variances for each feature in the the two classes. FLD values combine measuring this separation in means and variances, and are calculated using equation 2.27:

$$FLD = \frac{(\mu_1 - \mu_2)^2}{(\sigma_1^2 + \sigma_2^2)} \quad (2.27)$$

Within datasets, we note that the variance in the non-eating class is much higher than the variance in the eating-class. This is to be expected, as the non-eating class consists of any activity other than eating. For example, sedentary activities like resting or typing on a computer, or watching a movie, and non-sedentary activities like shopping, cooking food or folding laundry. These are just examples of possible non-eating activities. Since we do not have video evidence or other ground truth information, it is impossible to say.

Between datasets, we note that the variances for this new dataset are generally higher than the variances in [56], for example Acceleration(Eating), Roll(Eating), Roll(Non-Eating), and Roll Regularity(Non-Eating), while other features have stayed constant Manipulation(Eating), Acceleration (Non-Eating), Roll Regularity (Eating). These changes can be explained by the increase in motion at the wrist in the new dataset, facilitated by the low weight of the Shimmer3 (in comparison to the iPhone 4). Another explanation for the increase in variance is the amount of data in this new dataset, appx. 10 times more than [56], speaking to the breadth of variability captured by the new data collection. Only Manipulation(Non-Eating) shows a lower variation in this new dataset, and even this can be explained as a consequence of more linear motion at the wrist, a manipulation captures the ratio of wrist rotation to wrist linear movement.

We also note that the means for the acceleration feature are closer, and the FLD value lower in this new dataset than in [56]. One reason for this is the increase in the mean of sum\_acc for the

eating class in this new dataset, explained by the increase in wrist motion due to the lower weight of the Shimmer3. We hypothesize that the low weight of the Shimmer3 allowed participants to exhibit more free-living behavior than the iPhone did.

Table 2.3: Feature Values from previous work [56].

	Manipulation			Sum_Acc			Roll			Roll-Reg		
	Mean	Variance	FLD	Mean	Variance	FLD	Mean	Variance	FLD	Mean	Variance	FLD
Eating	791	45785	1.5	0.039	0.0002	0.05	9.1	18.2	0.09	0.58	0.02	0.49
Non-Eating	395	57284		0.054	0.0043		6.8	39.2		0.37	0.07	

Table 2.4: Feature values from this work.

	Manipulation			Sum_Acc			Roll			Roll-Reg		
	Mean	Variance	FLD	Mean	Variance	FLD	Mean	Variance	FLD	Mean	Variance	FLD
Eating	608	45808	0.84	0.057	0.0006	0.02	13.6	23.1	0.09	0.66	0.02	0.46
Non-Eating	337	41652		0.068	0.0043		9.5	65.4		0.45	0.08	

Table 2.5: Comparison of quantity of data between previous work and this new data collection.

Dataset	Subjects	Total (h)	Eating (h)	Meals
iPhone [56]	43	441	22	116
Shimmer3	351	4,680	229	1,133

Table 2.6: Time difference between logged meal start and end times, and nearest peaks in minutes

Dataset	Start Time Difference	End Time Difference
iPhone [56]	-0.6	1.5
Free-living	-4.5	7.3

Table 2.5 summarizes the differences in the number of subjects, hours of data collected, hours eating, and the total meals recorded. There is appx. 10x more data in the new Shimmer3 dataset compared to the previous iPhone dataset created by Dong et al. [56]. Table 2.6 evaluates the peak detection based segmenter. We learn that the time difference between peaks and self-reported start times is appx. 7 times higher in the new dataset compared to [56]. Similarly, the time difference between peaks and self-reported meal end times is appx. 5 times higher.

## 2.7.2 Weighted Accuracy for the Classifier

Table 2.7: Change in accuracy in replication experiment

Dataset	WACC	TPR	TNR
iPhone [56]	81	82	81
Shimmer3	75(-6)	71 (-11)	73 (-9)

Table 2.7 shows the results of testing the classifier using leave-one-out cross validation. The weighted accuracy per recording ranged from 26% - 99%, with the average accuracy per recording being  $74 \pm 13\%$ , with a median of 76%. The weighted accuracy for the dataset was 75%, at a TPR (recall) of 75% and a TNR (sensitivity) of 71%.

## 2.8 Conclusion

To our knowledge, this is the largest dataset known to us in the field of ADM, containing an order of magnitude more data in terms of subjects and hours than previous work (see table 2.8). We attempted to replicate previous work done by Dong et al. [56], with newer devices of smaller weight, and across a larger sample size. Compared to the weighted accuracy of 81% in previous

Table 2.8: Change in precision and recall of meal detection when transitioning from a controlled environment to free-living

Previous Work	Hours	Subjects	Ratio	Controlled		Free Living	
				Precision	Recall	Precision	Recall
Thomaz [71]	32	7	14:1	67	89	65 (-2)	79 (-10)
Mirtchouk [73]	245	11	12:1	88	87	45 (-43)	85 (-2)
Chun [57]	N/A	N/A	N/A	95	82	78 (-17)	73 (-9)
Zhang [74]	122	10	17:1	94	90	79 (-15)	77 (-13)
Dong [56]	449	43	20:1	N/A	N/A	N/A	82
<b>This Work</b>	4,680	351	20:1	N/A	N/A	N/A	71 (-11)

work, we see a weighted accuracy of 75% using a new wrist mounted device. This brings some confidence in the performance of the method, but also raises questions on why the performance of the method could not be replicated. Other authors have performed replication experiments comparing performance in controlled vs uncontrolled work. Table 2.8 compares previous work where experiments were replicated in controlled and free-living conditions. Previous work [57, 71, 73, 74] showed a drop in performance when experiments are replicated in free-living conditions. In these experiments, the drop in TPR (recall) ranges from 2% - 13%. While we did not attempt to replicate controlled experiments in uncontrolled environments, we still see a similar drop of 11% in TPR (recall). This drop can be explained by two changes in the replication experiment compared to the original experiment by Dong et al. [56].

Firstly, the device used for the new dataset was much less cumbersome than the iPhone mounted on the wrist, thus allowing more wrist activity. This can be seen in the increase of the mean value for the acceleration feature. During eating, the mean value increased by 46% between the iPhone and the Shimmer3 dataset, while increasing by 25% for the non-eating class. Thus we learn that the the form factor of a device used for detections in eating detection can greatly impact behavior. Consider other work where earbuds and piezo-electric necklaces are used as sensor platforms. Users not accustomed to eating while wearing these devices might not behave naturally during data collection. One way to address this issue may be through longitudinal data collection, where the subject can warm up to the device over the first few days, and then using data collected in later days for developing and testing methods. We consider this future work.

The second explanation for the drop in performance can be attributed to the larger number of subjects. We recruit an order of magnitude more subjects than any previous work, and include subjects from the general population in the cities of Clemson, Pendleton and Greenville. This



is in contrast to previous work where subjects are often students or staff on campus. While all subjects were compensated with \$25 cash, students and staff may be more motivated to comply with data collection protocol than the general population, as there may be implicit factors affecting the performance of students and staff recruited by researchers.

Another change in the results of the replication experiment was the time difference between peaks of wrist activity and meal boundaries, discussed in results table 2.6. One reason for this change could be a change in the signal used for peak detection - linear acceleration. The next chapter discusses estimating linear acceleration, and the presence of noise in its estimates.

## Chapter 3

# Filtering Linear Acceleration when Tracking Wrist Motion

Note: A significant portion of this chapter was published in the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) [104] and is ©IEEE.

The previous chapter described replication of the eating detection method, and showed that the performance drops when replicating the method on a larger subject pool while allowing for increased mobility of the wrist by using a smaller device with lesser weight. The method relies on features calculated from the linear acceleration of the wrist. Linear acceleration can provide rough estimates of the velocity and direction of wrist movement. It is estimated from raw acceleration (sensed by accelerators) by tracking sensor orientation and subtracting 1G in the direction of earth. However, it is well known that this calculation is prone to error [105]. Noise sources include errors in tracking sensor orientation, calibration bias due to variations in sensor manufacturing [106], and deviations of the magnitude of Earth’s gravity across the planet’s surface [107].

Noise in linear acceleration does not affect some applications of accelerometers such automobile crash testing [108] or pedestrian indoor navigation [109], as the acceleration being tracked is large compared to the noise. However, in applications tracking wrist motion, the low amplitude of wrist motion acceleration is affected due to the low signal to noise ratio. Features calculated from linear acceleration are more heavily affected by the noise present. Previous work has often avoided

estimating linear acceleration and instead used raw acceleration for features [110]. Others have evaluated low pass, high pass, bandpass or complimentary filters [99,100], recalibrated accelerometers in real-time [111], or modeled the gravity field [107]. These methods provide a general solution that is not specifically targeted at tracking wrist motion.

In attempting to explain the loss in accuracy when replicating Dong’s method of detecting eating [56], we found a change in the value of the sum of acceleration feature used for classification. Further exploration showed that estimations of linear acceleration from raw acceleration were impacted by sensor noise. This chapter describes the attitude heading reference system (AHRS) used to estimate orientation, and how this orientation is used to estimate gravity and linear acceleration. We then quantify linear acceleration and its noise, and describe a mean filter to mitigate the noise. By evaluating the performance of the eating detection algorithm on the iPhone data set and CAD, we show when to, and when not to make use of the mean filter.

## 3.1 Methods

The method to estimate linear acceleration noise is based on the assumption that during rest (zero velocity and zero acceleration), acceleration reported by a system (sensor or algorithm) must be noise. Therefore, noise in linear acceleration can be identified as the values reported by a system during periods of rest. Periods of rest can be identified as periods with low variance in angular velocity or acceleration over a small time window. Thus our rest detector work by calculating the statistical variance in angular velocity and acceleration.

This section first discusses the method to estimate linear acceleration. We then discuss the sources of noise that affect linear acceleration. Finally, we describe a rest detector to help estimate it, and a method to filter it.

### 3.1.1 Estimating Linear Acceleration

Microelectromechanical system (MEMS) accelerometers are constructed using spring-like objects that bend in a direction opposite to their motion [112], allowing them to sense the combined effect of acceleration and the gravity vector in their own frame of reference (relativistic acceleration). Previous work often modeled gravity as a low frequency signal, and thus used high pass or band pass filters to isolate linear acceleration [113]. In this work, we obtain linear acceleration  $a_l$  from

relativistic (raw) acceleration  $a_r$  by subtracting the gravity vector in the device frame  $g_d$ , as shown in Equation 3.1.

$$a_l = a_r - g_d \quad (3.1)$$

This gravity vector  $g_d$  can be obtained if the orientation of a device is known. Assuming the orientation is available in the form of a rotation matrix  $R$  of order  $3 \times 3$ , the gravity vector in the earth frame  $g_e = [0 \ 0 \ 1]g$  can be rotated to the gravity in the device frame  $g_d$ :

$$g_d = R g_e \quad (3.2)$$

### 3.1.2 Estimating Orientation (Pose)

When at rest, raw acceleration is equal to the gravity vector, and thus orientation (pose) can be obtained directly. However during motion, gravity needs to be separated from linear acceleration. Orientation can be tracked using dead reckoning, where angular displacement is obtained by integrating gyroscope angular velocity over time. However, the noise in gyroscope data causes drifts in these orientation estimates, making them unusable. Today, orientation is often tracked using a family of algorithms commonly known as AHRS (Attitude and Heading Reference Systems). Device manufacturers implement these algorithms in proprietary software such as Apple’s Core Motion library [114], or Invensense’s MotionProcessor API [115], while some open source sensor fusion implementations also exist [98]. Madgwick et. al recently introduced a complementary filter that fuses magnetic, angular rate and gravity (MARG) to estimate orientation [100]. This algorithm is preferred by researchers due to the availability of open source code implementations programmed in C, C# and Matlab. We hypothesize that proprietary implementations by Apple and Invensense implement variations of this algorithm, modified to work better with factory calibrated settings.

Madgwick’s algorithm uses gradient descent to provide orientation estimates in the form of quaternions. While rotation matrices are convenient and easily inferred by humans, quaternions are preferred in software as they are more compact and computationally efficient. Like a complementary filter, the algorithm defines the state of the system by a quaternion  $Q = q_0, q_1, q_2, q_3$  that represents the orientation. When new sensor input is available, an estimate of the new orientation  $Q_a$  is made using information from the accelerometer and magnetometer data. Another estimate of the

orientation  $Q_\omega$  is made by integrating the gyroscope angular velocity. The orientation at time  $t$ ,  $Q_t$  is then estimated from the quaternion at time  $t-1$ :  $Q_{t-1}$ , and the estimates  $Q_a$  and  $Q_\omega$ . Full details of this algorithm are provided in [100], while the source code is available at <https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/>.

The quaternion representing orientation can be converted to a rotation matrix using the equation below:

$$R = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}$$

Following this conversion, gravity can be obtained using equation 3.2, and then linear acceleration can be obtained using equation 3.1.

### 3.1.3 Noise in Linear Acceleration

While analyzing the collected data set we noticed plateaus in the magnitude of linear acceleration. These can be seen in figure 3.2. The range of values (0 to 0.6 m/s<sup>2</sup>), length of time, and flatness of change indicates they are not due to real motion and are instead noise artifacts. These plateaus are not noticeable during long periods of time, or if individual axis or short periods of motion (such as gestures) are visualized. Given that this noise exists after the raw acceleration has been processed by an AHRS algorithm, it is reasonable to believe that these errors are not corrected by such algorithms.

### 3.1.4 Sources of Error

Figure 3.1 shows an overview of the sources of error when calculating linear acceleration. Raw sensors (accelerometers, gyroscopes, and magnetometers) can all be affected by bias due to small offsets in coordinate systems or components during manufacturing [116]. Magnetometer readings can be distorted by local deviations in the magnetic field. Gyroscopes can be used to calculate orientation but the values must be integrated and thus suffer from drift in dead reckoning estimates. When an AHRS algorithm calculates object pose relative to the earth using this sensor data [112], it is affected by sensor noise. This pose  $R$  is then used to calculate gravity and linear acceleration

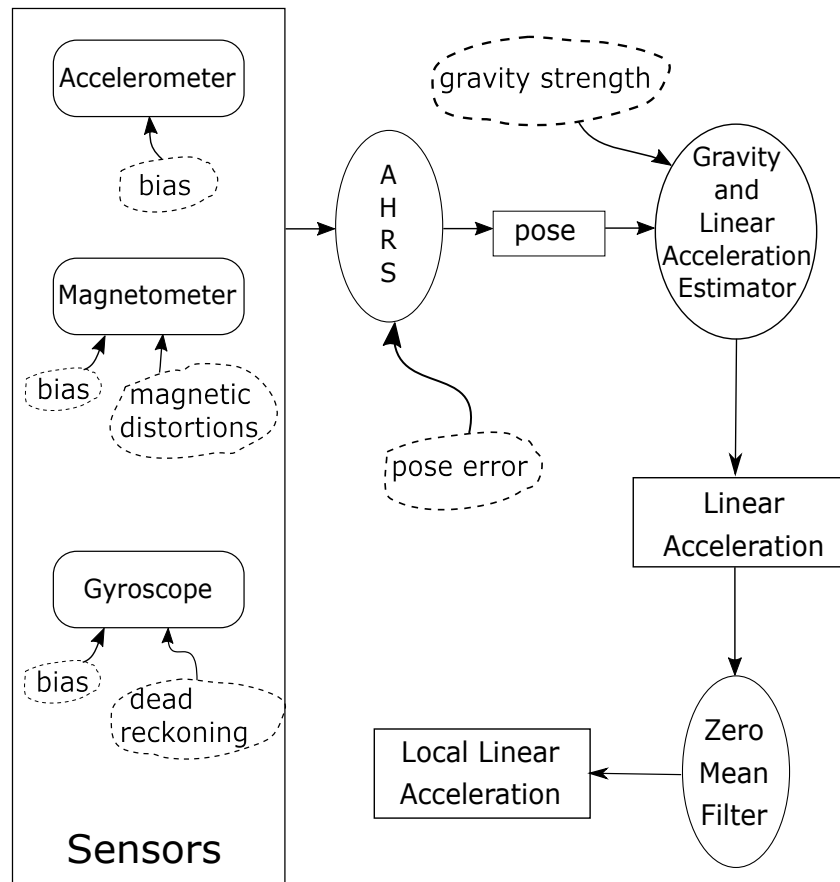


Figure 3.1: Source contributing to error in linear acceleration.

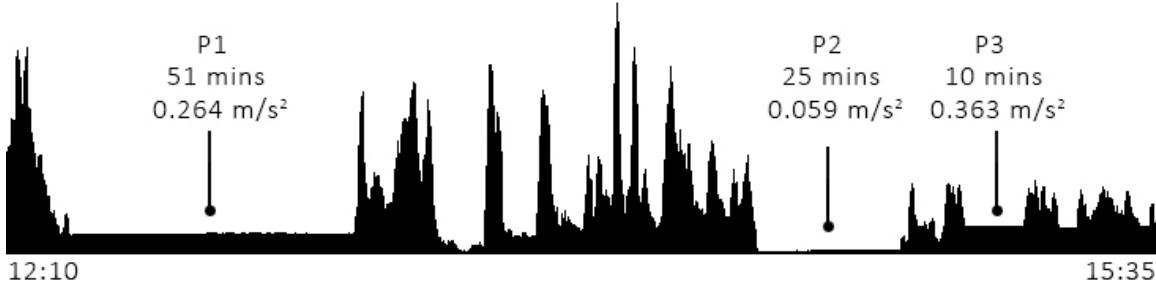


Figure 3.2: Magnitude of linear acceleration (Y-axis) vs time (X-axis). Noise can be seen as long plateaus of constant linear acceleration P1, P2 and P3.

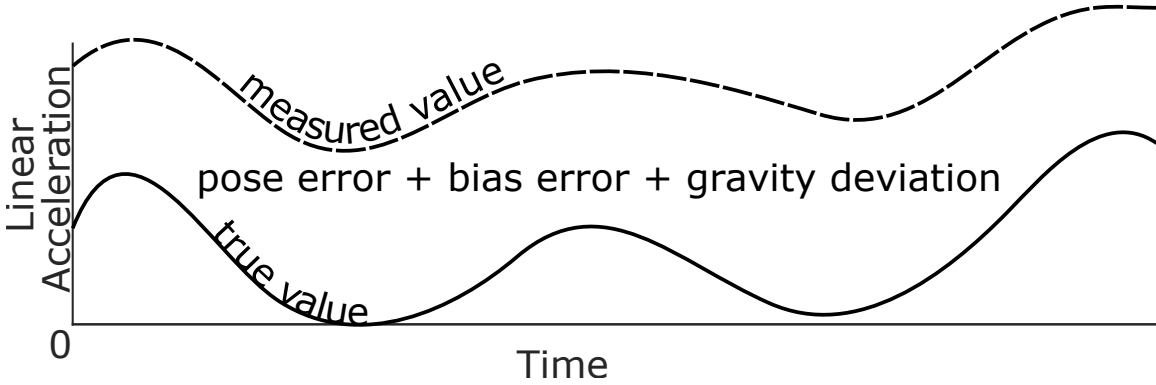


Figure 3.3: Graph showing an example of the composition of noise in linear acceleration during motion.

as shown in equation 3.2, estimates of which are also affected by this noise.

Errors in pose estimation are unavoidable [112] and even small errors can contribute to significant errors in linear acceleration. The earth gravity vector  $g_e$  has a standard value of  $1 \text{ g} = 9.81 \text{ m/s}^2$  but can vary from  $9.76 \text{ m/s}^2$  to  $9.83 \text{ m/s}^2$  for different locations on earth [117].

### 3.1.5 Rest Detector

When the device is moving, there is no known method of separating noise from the true value of acceleration. This is shown in figure 3.3. Noise can only be separated from acceleration signals when the device is at rest. When the device is not moving, we know linear acceleration  $a_l$  should have a true value of  $0g$ . All acceleration sensed can thus be attributed to noise (figure 3.4).

We used a variance based rest detector to mark datum as rest or motion for visualization. Variance was calculated for both accelerometer and gyroscope signals, thus checking for rest in wrist linear motion and rotation. For each datum at time index  $t$ , the standard deviation  $\sigma_t$  of each axis in

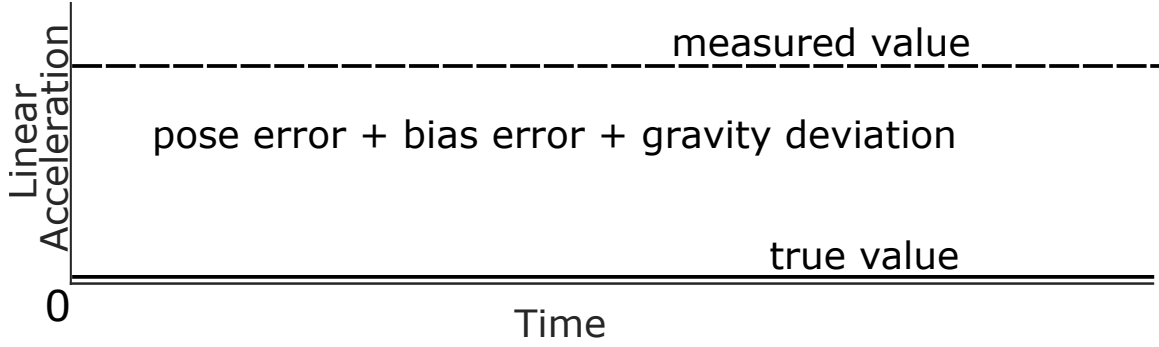


Figure 3.4: Graph showing an example of the composition of noise in linear acceleration during rest. As the expected acceleration is 0, any observed value of linear acceleration can be attributed to noise.

the acceleration  $\{A_x, A_y, A_z\}$  and gyroscope  $\{\omega_\phi, \omega_\psi, \omega_\theta\}$  signals was calculated over a fixed window centered at time index  $t$ . The sum of standard deviations of acceleration ( $\sigma_{A,t} = \sigma_{x,t} + \sigma_{y,t} + \sigma_{z,t}$ ) and gyroscope ( $\sigma_{\omega,t} = \sigma_{\phi,t} + \sigma_{\theta,t} + \sigma_{\psi,t}$ ) were then calculated. Time index  $t$  is assigned a state  $s_t = 0$  (rest) if  $\sigma_{A,t} < T_a$ , and  $\sigma_{\omega,t} < T_\omega$ , a state  $s_t = 1$  (motion) is assigned otherwise:

$$s(t) = \begin{cases} 1 & \text{if } \sigma_{A,t} < T_A \text{ and } \sigma_{\omega,t} < T_\omega \\ 0 & \text{otherwise} \end{cases}$$

The values  $T_a$  and  $T_\omega$  were tuned to 0.008g and 0.04 rad/sec by calculating the maximum value of standard deviation during segments visually identified as rest. The fixed window was tuned to 1 second. This allows the rest detector to detect periods of reasonable wrist rest, while avoiding being triggered by short moments where the sensor is not moving.

### 3.1.6 Zero Mean Filter

Noise in linear acceleration can be mitigated using a zero mean filter (ZMF). For each axis in the linear acceleration signal at each datum  $A_{i,t} = \{A_{x,t}, A_{y,t}, A_{z,t}\}$ , an average linear acceleration  $\bar{A}_{i,t}$  is calculated over a ten second window centered at the datum. This average is then subtracted from the value of linear acceleration to obtain corrected linear acceleration  $A'_{i,t}$ :

$$A'_{i,t} = A_{i,t} - \bar{A}_{i,t} \quad (3.3)$$

Figure 3.5 shows how the process changes when the mean filter is added to the eating



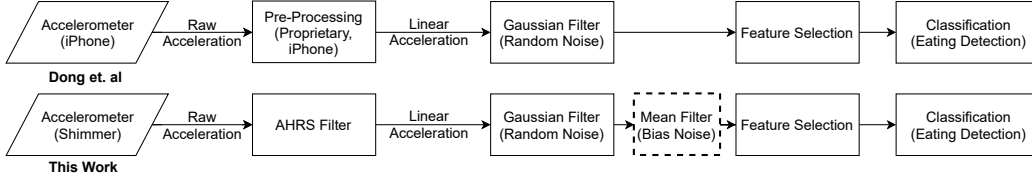


Figure 3.5: Flowchart showing the eating detection algorithm, and the addition of a mean filter to mitigate noise in linear acceleration before classification.

detection algorithm. The mean filter is added before the feature calculation step. It is important to note that this zero mean filter practically acts like a high pass filter. While information on the motion of the wrist relative to the body is retained, any information on global movements, like slow motions of the body while walking are filtered out.

## 3.2 Results

We analyzed linear acceleration data in CAD. Figure 3.6 shows the distribution of acceleration values during rest and motion. We learn that while some instances of high acceleration exist, the distribution of wrist linear acceleration is long tailed to the right. The distribution is concentrated in movements of relatively low acceleration. 50% of the acceleration is  $<0.01g$  (figure 3.6), 90% of the wrist acceleration is  $<0.04g$ , and 99.9% of the acceleration is  $<0.2g$ .

For the state of rest, linear acceleration values range from 0.00 to 0.06g, which can be interpreted as noise. Figure 3.6 shows that the average value for noise is in the neighborhood of linear acceleration when the wrist is in motion. More importantly, we learn that 70% of wrist motion lies within the range of noise, explaining the loss in performance in accuracy in CAD.

For some applications, this noise can be mitigated by using the mean filter described in section 3.1.6. Figure 3.7 shows the result on the magnitude of acceleration. While the noise during periods of rest is mitigated, signature motion signal is retained. The algorithm to automatically detect meals relies on linear acceleration for segmentation using a peak detector, and for calculating two features. When evaluating the algorithm on CAD, implementing the zero mean filter leads to an improvement of 1% in accuracy. On the other hand, implementing the filter leads to a 1% loss in accuracy on the iPhone dataset. Table 3.1 shows the change in specificity, sensitivity and accuracy for the two datasets on implementing the filter.

The filter improves the estimates of linear acceleration leading to better classification in

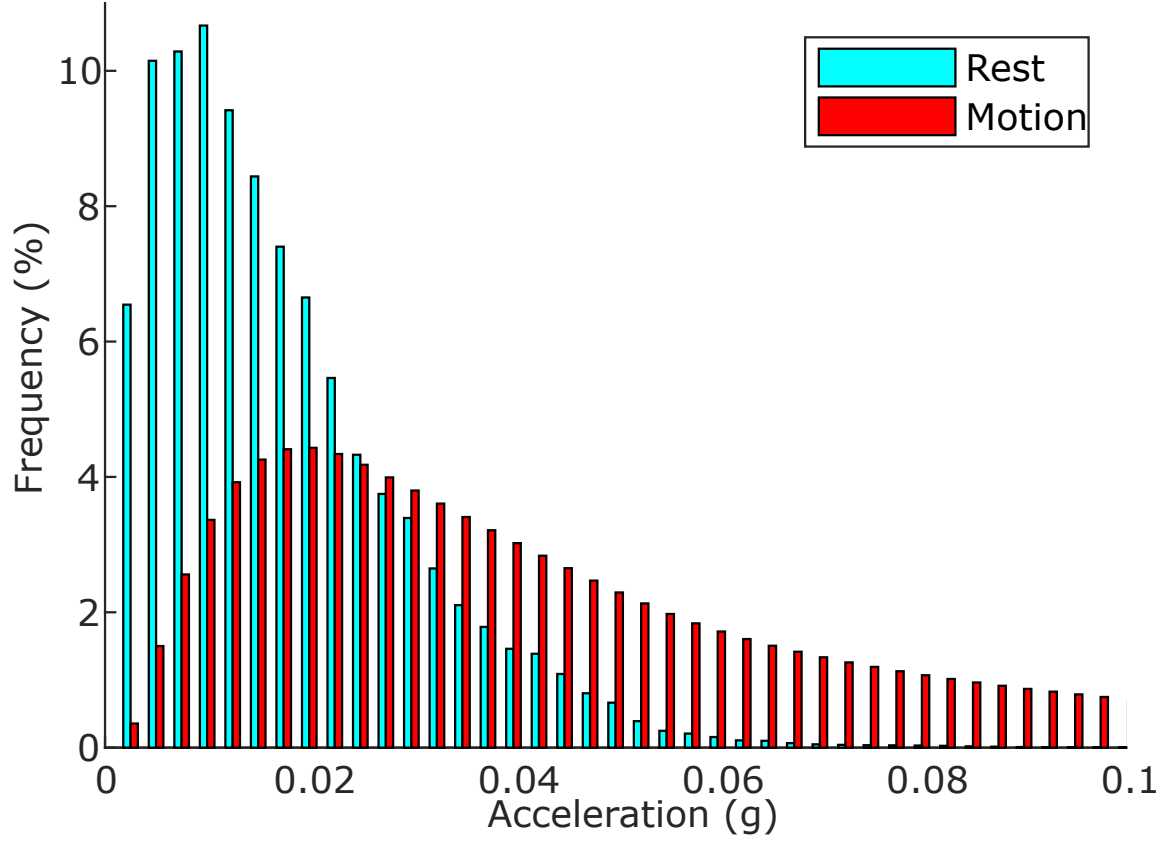


Figure 3.6: The spread of acceleration values for datum detect as rest (dark red) that can be interpreted as noise, motion (light blue). The overlap in the range of noise and signal is the motivation to use the mean filter.

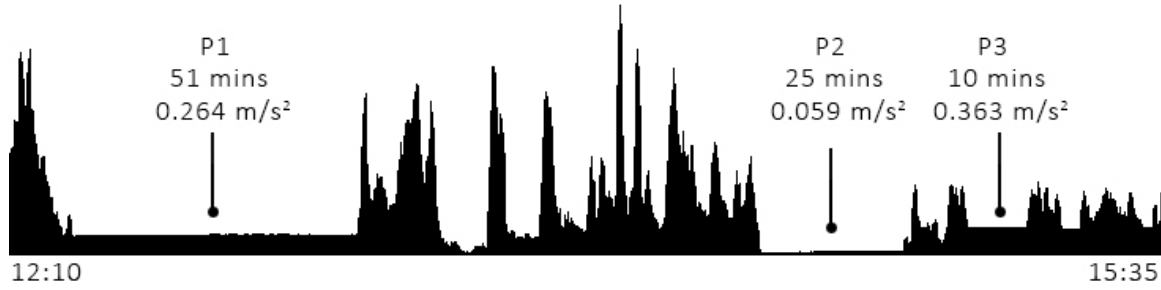


Figure 3.7: Magnitude of linear acceleration (Y-axis) vs time (X-axis). Plateaus P1, P2 and P3 in figure 3.2 are not seen after the mean filter is operated.

Table 3.1: Change of Specificity, Sensitivity, and Accuracy (%) in the meal detection algorithm after implementing the zero mean filter

Dataset	Initial	With Filter
Clemson All-day (CAD)	75/70/73	75/73/74
iPhone	78/83/80	76/83/79

CAD due to two reasons. The first: segmentation is improved as more peaks are detected. The hysteresis based peak detection algorithm performs better leading to better segmentation and thus classification. After implementing the filter, the average time between detected peaks was on average 8.5 seconds closer to the ground truth boundaries on CAD. On the iPhone dataset, the filter caused peaks to be missed and detected peaks 7 seconds further from the ground truth boundaries.

Second, better estimates of linear acceleration affect the manipulation feature value and its Gaussian distribution causing a larger separation in the classes. The Naïve Bayesian classifier is able to better model the non-eating and eating classes due to this separation.

### 3.3 Discussion

This chapter quantifies the difficulty imposed by noise sources in linear acceleration calculations. Noise in linear acceleration ranges from 0g to 0.06g, overlapping the range of wrist accelerations by 70%. The expected wrist acceleration is well within the range of noise, and thus linear acceleration are affected.

Noise in the linear acceleration estimates can be mitigated by using a mean filter. By evaluating the mean filter on a tracking based meal detection algorithm, we provide some evidence that applications that track wrist motion may benefit from the zero mean based noise filter. The method improves classification accuracy on data collected using the Shimmer3, but shows a loss in data collected by the iPhone4. One reason for this could be the iPhone4's proprietary code, which may already filter linear acceleration. Applying a (second) zero mean filter causes too much dampening thus leading to loss of information in the signal.

When used correctly, this filter mitigates noise while preserving the general trends that indicate motion. The practical effect of this filter is that local motion (wrist relative to the body) is maintained, at the cost of information on global motion (e.g. distance walked by a subject wearing an accelerometer).

## Chapter 4

# The Impact of Walking and Resting on Automated Detection of Meals

Note: A significant portion of this chapter was published in the ACM Transactions on Computing for Healthcare (HEALTH) [95] and is ©ACM.

In section 1.9, we had discussed how researchers often evaluate methods on data collected in the lab, as collecting data from free-living subjects is complicated. Recent work in automated eating activity recognition has shown that classifier accuracy decreases when transitioning from the laboratory to free living (see table 4.1), however there is very little published discussion on it's cause.

Thomaz et. al showed that in a laboratory setting activities like chatting, using the phone, and brushing hair with a comb can be confused with eating [71]. Zhang et. al [118] learned that most meals are consumed when a human is stationary, and thus excluded periods of walking showed that walking often looks like eating gestures in free-living. We hypothesize that another important

Table 4.1: Change in precision and recall of meal detection when transitioning from a controlled environment to free-living

Previous Work	Controlled		Free Living	
	Precision	Recall	Precision	Recall
Thomaz [71]	67	89	65 (-2)	79 (-10)
Mirtchouk [73]	88	87	45 (-43)	85 (-2)
Chun [57]	95	82	78 (-17)	73 (-9)
Zhang [74]	94	90	79 (-15)	77 (-13)

factor may be secondary activities conducted concurrently with eating. Laboratory tests have advantages in that the data can be collected under direct observation which makes the annotation of ground truth behaviors more simple, and because a scripted list of activities can be given as instructions to subjects. However, while being directly observed, subjects are unlikely to conduct secondary activities while eating unless specifically instructed to do so. In contrast, collecting data from free-living subjects is more complicated [57, 71, 73] because tools that can be used in the lab like a video camera cannot be easily implemented in a free-living setting, and when they are used, they bring concerns about the privacy of the subject and people they interact with [83]. Participants may also exhibit behaviors in free-living that were not captured in the lab, as they are often limited by tracking devices utilized in the lab or when under observation. For example, Alharbi et. al [82] note how subjects wearing cameras during the collection of eating activity data experience social and surveillance discomfort. Mirtchouk et. al [73] note that their classifier performed poorly on one subject out of eleven (9%) as that subject had extended conversations and multi-tasked (did homework) while eating a meal. A recent study performed statistical tests on data collected from participants in a lab versus data collected from participants in a free-living facility, and reported numerous differences including changes in the number of bites, the time spent eating, and the time and number of pauses between ingestion events [81].

In this chapter, we discuss the presence of secondary activities, and how they affect the detection of meals in free-living subjects. First, we analyze the information on secondary activities conducted during eating, collected for the Clemson All-day data set. We quantify how often these activities occur in free-living. Second, we show secondary activities are often multitasking activities. This multitasking leads to overlapping and intermittent secondary activities, causing difficulty modeling such meals. Third, we develop a classifier to detect two common secondary activities: walking and resting. These two activities exhibit motion patterns that are distinctive enough from eating that they can be reliably detected as sub-activities within periods of time that subjects report as meals or snacks. Combined, these results help explain why accuracy drops when methods to detect eating are tested in free-living, and how detecting secondary activities is key to improved detection of periods of eating.

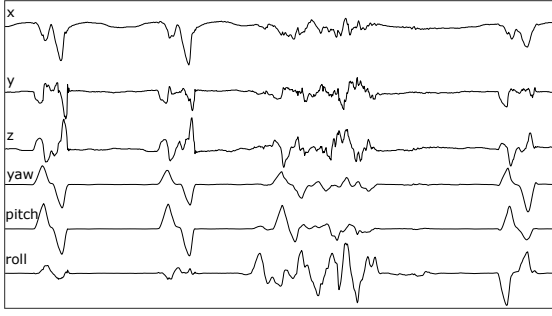


Figure 4.1: Example of one minute of wrist data (linear accelerations  $x$ ,  $y$ ,  $z$  and gyroscope yaw, pitch, roll) from a person eating a banana with rest between bites.

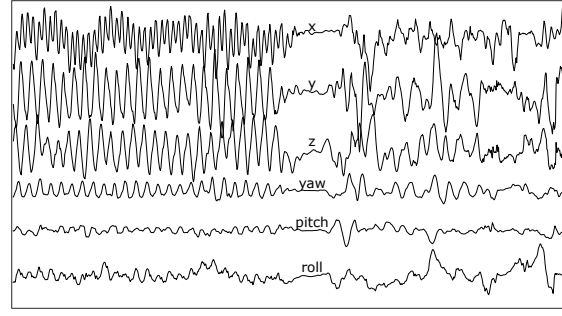


Figure 4.2: Example of one minute of wrist data from a person walking regularly for 30 seconds, followed by a short stop, then looking around a room.

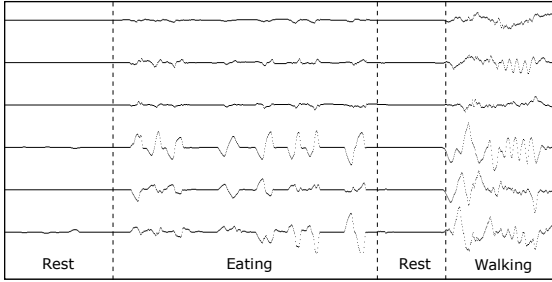


Figure 4.3: Wrist data from the end of a meal, showing intermittent eating, walking and resting. In this segment, the subject rested for a period of 15 seconds, consumed food for 35 seconds, rested briefly, then walked for 15 seconds before ending the meal.

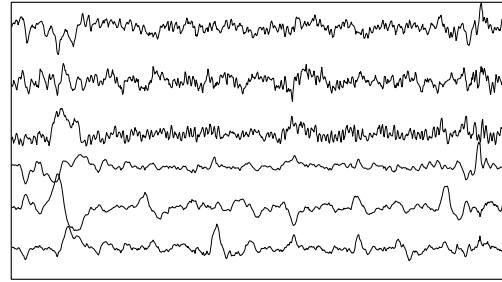


Figure 4.4: Example of one minute of wrist data (linear accelerations from a person eating a banana while walking). The secondary activity of walking obfuscates the wrist motion signals indicative of food consumption.

## 4.1 Overview

We first demonstrate how wrist motion patterns associated with eating can be obfuscated by wrist motions associated with secondary activities. Figure 4.1 shows 60 seconds of wrist motion data from a person eating a banana. Top to bottom are accelerometer  $x$ ,  $y$ , and  $z$ , gyroscope yaw, pitch, and roll. Modulations in the signals are caused by wrist motions moving food to the mouth and then moving the wrist back to a neutral position, while periods of no motion indicate when the wrist was at rest. This type of motion is typical during consumption. Figure 4.2 shows 60 seconds of wrist motion data from a person walking down a hallway and then entering a room and searching around inside it. Swinging the arms during walking causes sensor patterns that have regular oscillations and look clearly different than the sensor patterns during eating.

One can imagine several scenarios in which a person might walk or rest during a meal or

snack. For example, a person might walk around a kitchen to prepare a second serving of food before sitting down and resuming consumption. A person might rest for several minutes while watching TV before resuming consumption. Figure 4.3 shows an example of this type of behavior towards the end of a meal in which a person rested for 15 seconds, engaged in consumption for 35 seconds, briefly rested, and then walked for 15 seconds before indicating the end of their meal. These secondary activities occurred intermittently within the period of eating, but depending on their frequency and duration they can greatly increase the difficulty of detecting the period of eating. Figure 4.4 shows one minute of wrist motion data from a person eating a banana while walking. In this case the secondary activity is occurring concurrently during eating, which obfuscates all the wrist motions associated with eating.

Classification of free-living meals containing secondary activities could be performed by modeling mixtures of activities as different classes. However, a secondary activity may not be conducted continuously for the entire duration of the eating activity. We therefore take an approach where subsegments of a self reported meal are analyzed and classified independently. We specifically consider two secondary activities: walking and resting. We present two experiments. In our first experiment, we develop detectors for periods of walking and for periods of resting. We test these detectors on two data sets for which video ground truth of activities is available. The first data set was collected for a pedometer experiment and is known via video to contain 100% walking. The second data set was collected in a cafeteria and via video is known to contain 6.8% rest during eating episodes. In our second experiment, we use the walking and resting detectors to measure how frequently these secondary activities occur during periods of eating in free-living. No video-based ground truth is available for the second experiment, so we rely upon the results of the first experiment to provide confidence in the measures found in the second experiment. Finally, we augment a previous existing eating detection algorithm with the walking and resting detectors and measure their impact on detecting periods of eating in the free-living data set.

## 4.2 Walking and Resting Detection

The purpose of this experiment is to develop classifiers for detecting walking and resting. We test them on datasets that have video-based ground truth of subject activities. The goal is to provide confidence that they work reliably enough to detect walking and resting on additional

datasets for which video-based ground truth is not available.

#### 4.2.1 Datasets

A dataset containing wrist activity data during walking was collected by Mattfeld et. al [119] for pedometer algorithm evaluation. A total of 30 subjects were recorded. Each subject was instrumented with three Shimmer3 sensors (wrist, hip, and foot) and was followed by an experimenter using a smartphone to record synchronized video of their lower body. Subjects walked an outside path, inside a building, and inside a room, collectively taking over 60,000 steps. For this work, we use the data collected from the wrist while the participant walked an outside path. The dataset is publicly available at <http://www.cecas.clemson.edu/~ahoover/pedometer>.

A second data set was collected in a cafeteria setting by Shen et. al [54]. A total of 271 subjects were recorded, each consuming a single meal. Subjects sat at an instrumented table that had video cameras installed above it in the ceiling to record each subject and their food while they ate. Each participant wore a custom device on the wrist housing MEMS accelerometers and gyroscopes. Ground truth of bites and other eating-related gestures like resting and manipulating food was provided by trained reviewers watching the synchronized video. The data set is publicly available at <http://www.cecas.clemson.edu/~ahoover/cafeteria>.

#### 4.2.2 Preprocessing and Segmentation

All the data sets used in this work were recorded at 15 Hz. To reduce sampling noise, we filter raw acceleration and gyroscope signals  $R_t = \{a_x, a_y, a_z, \omega_\phi, \omega_\theta, \omega_\psi\}$  at time index  $t$  to smoothed signals  $S_t = \{S_{x,t}, S_{y,t}, S_{z,t}, S_{\phi,t}, S_{\theta,t}, S_{\psi,t}\}$  using a standard Gaussian filter [120] operated independently on each axis. The filter operates on a window of past data one second long using a Gaussian  $\sigma$  of 10 s.

For classification, we segmented the data into fixed one minute windows, starting 10 seconds before the first step (pedometer dataset) or bite (cafeteria dataset), and ending 10 seconds after the last step or bite. Segments smaller than one minute were discarded.



## 4.2.3 Detection of Secondary Activities

### 4.2.3.1 Walking

A zero-crossing based algorithm is employed to detect walking. For a given segment, a datum is identified as a zero-crossing  $z(t) = 1$  if any axis ( $S_{any,t}$ ) in the gyroscope signal ( $S_{\phi,t}$ ,  $S_{\theta,t}$ ,  $S_{\psi,t}$ ) crossed zero (from negative to positive or vice-versa). To suppress spurious detection of zero-crossings from noise, the signal must also surpass  $\pm 5$  deg/sec in the direction of the zero crossing. The threshold of  $\pm 5$  deg/sec was decided heuristically to avoid spurious detection of zero crossings caused by noise in the signal during rest but is small enough to detect the oscillations of the wrist caused by walking.

$$z(t) = \begin{cases} 1 & \text{if } \text{sgn}(S_{any,t-1}) \neq \text{sgn}(S_{any,t}) \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

The feature  $f_{1,w}$  provides the rate of zero crossings for a segment  $w$  of length  $W$  and is calculated using equation 4.2. It defines the percent of data in the segment identified as zero crossings, and ranges from 0 to 1. Larger values (large amounts of zero crossings) tend to occur during walking compared to other activities. A segment is considered walking if  $f_{1,w}$  is greater than threshold  $T_1$ .

$$f_{1,w} = \frac{1}{W} \int_W z(t) \quad (4.2)$$

### 4.2.3.2 Resting

Resting can be detected by looking for low variance in accelerometer and gyroscope signals. By using variance in the signal, we are able to detect rest regardless of the orientation of the device. For robustness, our classifier uses two steps. Each datum is first classified as rest  $r(t) = 1$  or motion  $r(t) = 0$ . To do this the standard deviation  $\sigma_t$  is calculated over a window of  $M = 1$ s for each of the six signals  $S_{x,t}$ ,  $S_{y,t}$ ,  $S_{z,t}$ ,  $S_{\phi,t}$ ,  $S_{\theta,t}$ ,  $S_{\psi,t}$ . If the sums of standard deviations for the acceleration ( $\sigma_{A,t} = \sigma_{x,t} + \sigma_{y,t} + \sigma_{z,t}$ ) and gyroscope ( $\sigma_{\omega,t} = \sigma_{\phi,t} + \sigma_{\theta,t} + \sigma_{\psi,t}$ ) signals are less than  $T_A$  and  $T_\omega$  respectively, the datum is considered to be at rest:

$$r(t) = \begin{cases} 1 & \text{if } \sigma_{A,M} < T_A \text{ and } \sigma_{\omega,M} < T_\omega \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

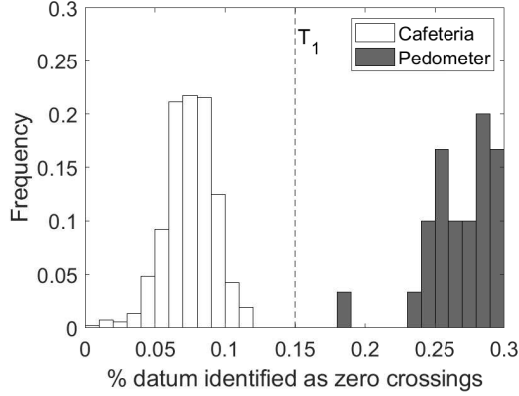


Figure 4.5: Histogram showing amount of zero-crossings per subject in the pedometer and cafeteria datasets.

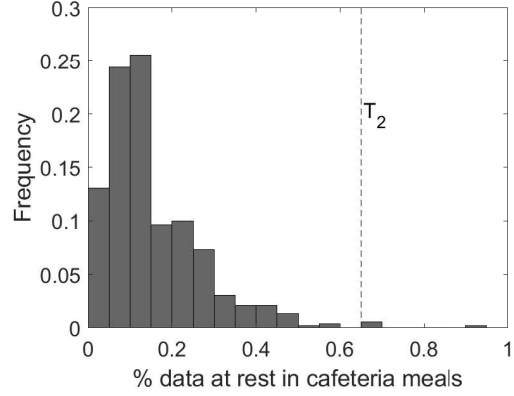


Figure 4.6: Histogram showing the amount of rest per meal in the cafeteria dataset.

A segment is then classified as rest if the percentage of data labeled rest in the segment is greater than the threshold  $T_2$ . The feature  $f_{2,w}$  (equation 4.4) ranges from 0 to 1. Larger values tend to occur during resting compared to other activities.

$$f_{2,w} = \frac{1}{W} \int_W r(t) \quad (4.4)$$

#### 4.2.4 Parameter Tuning

Parameters for the walking and resting detectors were tuned using histogram analysis of the pedometer and cafeteria datasets. Figure 4.5 shows the values of  $f_{1,w}$  for data from each participant in the datasets. We set  $T_1 = 0.15$  in the middle of the two histograms to label all walking segments as walking, and all cafeteria meals as not walking, since the cafeteria dataset did not contain any periods of walking.

For the rest detector,  $T_A$  and  $T_w$  were set by calculating the maximum value of standard deviation in the acceleration and gyroscope signals in segments visually identified as rest. To identify threshold for the amount of rest in a segment ( $T_2$ ), we plotted a histogram of the amount of rest in cafeteria meals (figure 4.6).  $T_2$  was set to 0.65 based on this histogram. There were some meals in the cafeteria where the amount of rest was greater than 65%, however video evidence showed the subject was eating with the non-instrumented hand, with the instrumented hand largely at rest.

## 4.3 Free-living Eating Activities

In this section we discuss free-living eating activities and their classification. We describe the collection of a large new data set and a classifier to detect periods of eating.

### 4.3.1 Clemson All-day data set (CAD)

The Clemson University Institutional Review Board approved the collection of this data set. Participants were recruited from the student body, faculty, staff, and residents of the surrounding area. Each subject collected data for one day during free living. Prior to the day of recording, subjects met with an experimenter that trained them on how to use the Shimmer3 device. Subjects were instructed to put on the device upon waking, turn it on to start recording data, and to take the device off prior to going to sleep at the end of the day. Throughout the day they were instructed to press a timestamp button at the beginning and end of any period where the primary activity was consumption, like a meal, snack or other contiguous period of eating. Subjects were instructed not to timestamp grazing consumption activities, or activities where consumption was not the primary activity such as drinking a cup of coffee over a long period of time while working or reading a book. Subsequent to data collection, subjects met with an experimenter for an exit interview. Logs of the timestamps of device button presses were reviewed with subjects to help identify erroneous button presses and to pair start and stop times of eating. The reviews were also used to collect secondary information including location (e.g. home or restaurant), type (e.g. lunch or dinner), if the eating activity consisted of multiple servings of food (yes or no), if the participant was eating in company (yes or no), what was consumed (open response), and if any secondary activities were being performed while eating (open response).

A total of 408 subjects were recorded (61% female, BMI  $25.8 \pm 5.8 \frac{kg}{m^2}$ , age  $28 \pm 12$  years). Data was collected on all seven days of the week. Of these, 351 (86%) completed a recording that was usable. Recordings were unusable for the following reasons: 19 device failures, 4 people forgot to turn it on, 11 people took it off partway through the day, 2 people carried it in a pocket instead of wearing it, 2 devices were lost or damaged, and 13 people failed to follow instructions. One subject recorded for three days and another subject recorded for two days, yielding a total of 354 days of usable data. The total duration recorded was 4,680 hours, containing 229 total hours of eating across 1,133 separate eating activities (meals, snacks). The average duration recorded per subject was 13.2

hours.

### 4.3.2 Preprocessing

Our algorithm uses features based on linear acceleration which is calculated from accelerometer values by subtracting gravity in the orientation of the device. A proprietary attitude heading reference system (AHRS) algorithm operating inside the Shimmer3 reports the orientation of the device as a quaternion  $Q = q_0, q_1, q_2, q_3$ . We convert this quaternion to a rotation matrix  $R$  using the equation below.

$$R = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}$$

Assuming a gravity vector  $G_e$  of  $[0 \ 0 \ 1]g$  in the earth frame, gravity  $G$  in the device frame can be obtained from the rotation matrix  $R$  and as:

$$G = [R \cdot G_e] \quad (4.5)$$

Linear acceleration can thus be obtained from the raw acceleration and gravity in device frame as:

$$l = A - G \quad (4.6)$$

where  $A$  is the raw acceleration  $[a_x, a_y, a_z]$ , and  $G$  is the gravity vector in the device frame. Linear acceleration can be noisy due to errors in orientation tracking. We apply a high pass filter by subtracting the average linear acceleration value over a sliding 1 minute window from each datum.

Raw signals  $R_t = \{l_x, l_y, l_z, \omega_\phi, \omega_\theta, \omega_\psi\}$  at time index  $t$  are filtered using a standard Gaussian kernel with the same parameters as those described in section 4.2.2. The result is smoothed signals  $S_t = \{S_{x,t}, S_{y,t}, S_{z,t}, S_{\phi,t}, S_{\theta,t}, S_{\psi,t}\}$ .

### 4.3.3 Segmentation

Previous work by our group has shown that wrist motion activity peaks before and after meals with peaks being in temporal proximity to self-reported meal start and end times [56]. It is

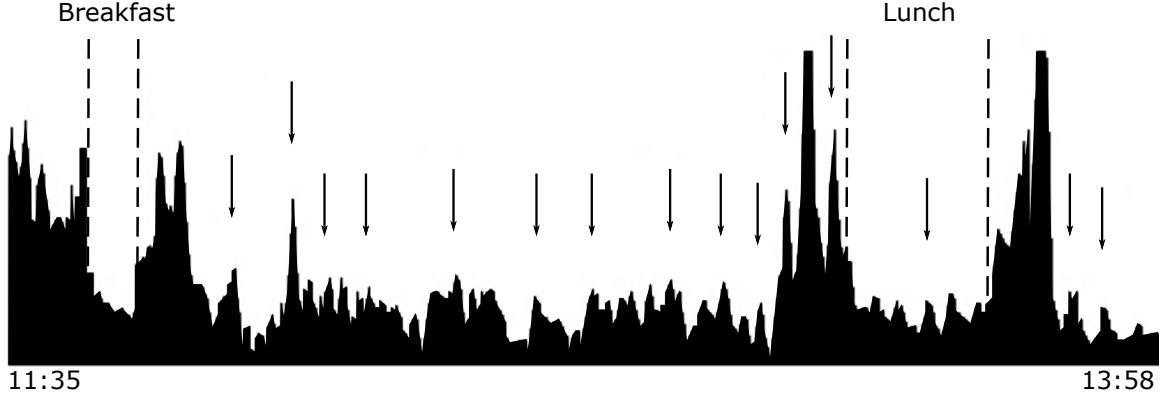


Figure 4.7: Wrist motion activity during a day. Automatically detected peaks used to segment the data for classification are indicated with arrows. Self-reported eating activities are marked with dashed lines.

assumed that these peaks are caused by actions before an eating activity commonly related to meal preparation, and those after eating related to cleaning up. Figure 4.7 shows an example of the same pattern seen in our new data collection. Wrist motion activity (magnitude of acceleration) is plotted on the Y-axis versus time on the X-axis.

Data is segmented as periods between peaks. We use the same method for peak detection described in [56], which uses a hysteresis approach to find local maxima in the sum of linear acceleration. Peaks detected by this algorithm are marked by arrows in figure 4.7.

#### 4.3.4 Classification Features

Peak to peak segments are classified as walking, resting, eating or other. Six features are calculated for each peak-to-peak segment. Two features characterize resting and walking, while four features characterize eating and were first introduced in [56]. The first feature  $f_1$ , the rate of zero crossings, and  $f_2$ , the amount of rest in a segment are described in section 4.2.

The third feature  $f_3$  is called manipulation, and measures the ratio of wrist rotation to linear motion:

$$f_{3,w} = \frac{1}{W} \sum \frac{|S_{\phi,t}| + |S_{\theta,t}| + |S_{\psi,t}|}{|S_{x,t}| + |S_{y,t}| + |S_{z,t}|} \quad (4.7)$$

where  $f_{3,w}$  is the value of the manipulation feature for the segment with time span  $W$  (number of samples),  $S_{x,t}$ ,  $S_{y,t}$ ,  $S_{z,t}$  are the smoothed linear acceleration values for the respective axes, and  $S_{\phi,t}$ ,  $S_{\theta,t}$ ,  $S_{\psi,t}$  are the smoothed angular velocities (yaw, pitch, roll) from the gyroscope. The fourth

feature, linear acceleration, is calculated as:

$$f_{4,w} = \frac{1}{W} \sum^W |S_{x,t}| + |S_{y,t}| + |S_{z,t}| \quad (4.8)$$

Wrist roll motion is calculated as:

$$f_{5,w} = \frac{1}{W} \sum^W |S_{\phi,t} - \frac{1}{W} \sum^W S_{\phi,t}| \quad (4.9)$$

Regularity of wrist roll motion is calculated as:

$$f_{6,w} = \frac{1}{W} \int_W 1 \ \forall \ t \in [ |S_{\phi,t}| > 10^\circ \dots t + 8\text{sec} ] \quad (4.10)$$

This feature represents the percentage of time the wrist was rolling, and takes a value between 0 and 1. This time is calculated as the amount of time the wrist roll was at least 10 deg/sec, plus the next 8 seconds after the wrist roll reduces to less than 10 deg/sec. The values 8 sec and 10 deg/sec were tuned in [56].

### 4.3.5 Classification

We use a two-stage classifier. In the first stage, thresholds determine if a segment is walking or resting. In the second stage, remaining segments are classified using a Bayesian classifier as eating or other. A segment is considered as walking if  $f_{1,w} \geq T_1$ . Similarly a segment is considered to be resting if  $f_{2,w} \geq T_2$ . The nature of peak-to-peak segmentation allows this method to only label segments as walking or resting if they are sufficiently long and largely walking or resting.

Segments not considered walking or resting are then labeled as eating or other by a Naïve Bayesian classifier that assumes independence of features. This classifier assigns a class  $c_i \in C$  to a segment given feature values  $f_j$  as shown in equation 4.11.

$$c_i = \arg \max_c P(c_i) \prod_j P(f_j | c_i) \quad (4.11)$$

We have only two classes for the Bayesian classifier, eating ( $c_3$ ) and other ( $c_4$ ). We tested different values of prior probabilities  $P(c_3)$  and  $P(c_4)$ , and found the best balance between eating and non-eating detection at  $P(c_3) = P(c_4) = 0.5$ . A normal distribution was used to calculate the

probabilities of segment belonging to these classes given their feature values as shown in equation 4.12:

$$P(f_j|c_i) = \frac{1}{\sqrt{2\pi\sigma_{i,j}^2}} \exp\left(-\frac{(f_j - \mu_{i,j})^2}{2\sigma_{i,j}^2}\right) \quad (4.12)$$

where  $\mu_{i,j}$  is the mean of feature  $j$  in class  $i$  and  $\sigma_{i,j}^2$  is the variance. Collectively, the two-stage classifier can be stated as equation 4.13:

$$c_i = \begin{cases} 1 & \text{if } f_{1,W} \geq T_1 \\ 2 & \text{if } f_{2,W} \geq T_2 \\ \underset{c}{\operatorname{argmax}} P(c_i) \prod_{j=3}^6 P(f_j|c_i) & \text{otherwise} \end{cases} \quad (4.13)$$

### 4.3.6 Parameter Tuning

Eating activity classification uses normal distributions trained from the free-living data sets. We use standard leave one out cross validation for training and testing. For feature values of the other class, data labeled other was split into five minute segments and feature values were calculated. For the eating class, feature values were calculated for time periods between the self-reported meal start and end times.

### 4.3.7 Evaluation Metrics

We use two metrics to evaluate our methods: an activity level recall, and a per-second metric. The activity level recall evaluates how many self-reported eating activities intersected with a machine detected meal, and answers the question ‘‘Can eating activities be detected?’’. The per-second metrics evaluate how many seconds of eating were correctly classified, answering the question ‘‘Can we detect the time period during which someone is eating?’’.

Precision, the F1 score and Cohen’s Kappa are well known to be affected by class imbalance [89, 90, 121]. Eating occurs far less frequently than not eating, so that the detection of periods of eating is by default an imbalanced problem [56]. We therefore evaluate our classifier using weighted accuracy [56, 73] to accommodate the imbalance in occurrence of eating vs other:

$$WACC = \frac{TP \times 20 + TN}{P \times 20 + N} \quad (4.14)$$

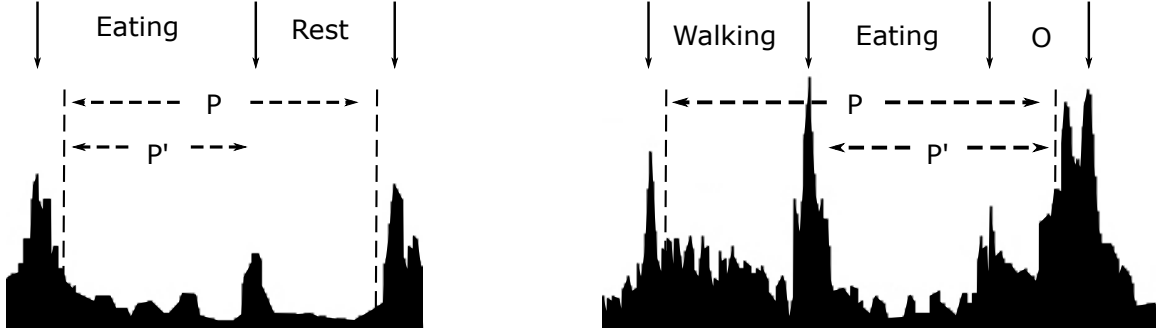


Figure 4.8: Wrist motion data examples. Peaks are marked with arrows and self-reported meal boundaries are marked as dashes. Machine detected labels are indicated between segments, with O indicating other. P contains all data in a self-reported segment, while P' excludes segments identified as walking or resting.

where  $WACC$  is the weighted accuracy, P (positives) are the total number of seconds in self-reported meals, N (negatives) are the total seconds labeled non-eating, TP (True Positives) are the number of seconds classified as eating inside self-reported meal times, and TN (True Negatives) are the number of seconds classified as other outside self-reported meal times.

In the above equation, P (positives) includes secondary activities that may have occurred during a self-reported meal. When these secondary activities are detected correctly as non-eating, they are labeled as false negatives. Consider the two example meals shown in figure 4.8. While walking and resting are correctly detected, equation 4.14 penalizes the classifier for not labeling some segments as eating that were in fact not eating. We adjust for this by using P' instead of P:

$$WACC' = \frac{TP \times 20 + TN}{P' \times 20 + N} \quad (4.15)$$

where P' is the number of seconds in self-reported meals with walking and resting removed i.e.

$P' = P - (\text{Walking} + \text{Resting})$ , and  $WACC'$  is the adjusted weighted accuracy.

## 4.4 Results

In this section we first provide evidence on the prevalence of secondary activities in free-living meals. We then show the performance of the walking and resting detectors on the pedometer (walking) and cafeteria (eating) datasets for which ground truth video is available. Having confidence on the performance of these detectors, we show how much walking and resting is seen during



Table 4.2: Percentage of walking and resting detected during periods of walking in the Pedometer dataset, and during eating in the Cafeteria datasets and Free-living datasets.

Dataset	Total data (hrs)	Eating (hrs)	% duration detected resting	% duration detected walking
Pedometer	4.7	0	0%	100%
Cafeteria	96	96	7%	0.3%
Free living	4,680	239	12.8%	5.5%

free-living eating (table 4.2). We discuss the performance of the classification algorithm, and how adjusting for secondary activities like walking and resting affects its accuracy.

#### 4.4.1 Walking and Resting

We evaluated the walking and resting classifiers on one minute segments in the pedometer and cafeteria datasets. These datasets contain 4.7 hours of walking (pedometer) and 96 hours (cafeteria) of data collected from 30 (pedometer) and 271 (cafeteria) subjects. In the pedometer data, all one minute segments (100%) were correctly classified as walking. In the cafeteria data, 0.3% of one minute segments were classified as walking. The rest detector detected no one minute segments as rest in the pedometer dataset, while labeling 7% of the one minute segments in the cafeteria dataset as rest. Video evidence from the cafeteria dataset revealed rest 6.8% of the time. Both the detectors have a slight false positive rate. We conclude that the classifiers are reliable enough to be used on the free-living dataset in our second experiment.

In the free-living data set, we found that subjects walked 5.5% of the time during meals. Walking virtually never happens in controlled laboratory experiments involving eating. If a classifier was trained on laboratory data and then deployed to free-living, it could be expected that the presence of walking would be new and likely to reduce accuracy. In the free-living data set, we found that subjects rested 12.8% of the time during meals. In the cafeteria data set resting was 7%. It is likely that subjects take longer and rest more during free-living than in a controlled setting because they are conducting passive secondary activities such as watching television. This again could help explain why a classifier trained on laboratory data might have lower accuracy when tested on free-living data.

Table 4.3: Some of the activities reported during a meal.

Putting away groceries	Playing with niece	Walking dog
Watching media on computer	Sitting in meeting	House chores
Checking email and internet	Working on papers	Reading book
Talking and playing pool	Sitting in car	Talking in car
Talking to team in class	Getting ready	Doing work
Talking to friend in a fair	Playing trivia	Working
Talking and texting	Watching TV	Reading

#### 4.4.2 Secondary Activities in Free-Living Eating

Of a total 1,133 meals self-reported in the new data set, 317 (28%) were consumed without any secondary activity, 281 (25%) were consumed while talking to company, and 535 (47%) were consumed while performing a secondary activity. A short list of some of the activities is shown in Table 4.3. Of note are activities during eating previously unseen in the literature like putting away groceries, playing with niece, walking dog, or house chores. This list demonstrates the breadth of complexity of the free-living dataset in representing free-living behavior.

Of the 1,133 self-reported eating activities, 21 (1.6%) were classified as walking and 38 (3.4%) were classified as resting. After examining each of these meals individually, we concluded that secondary activities dominated the wrist motions so much that consumption could not be seen. We therefore excluded them from the following analysis. We also excluded 9 meals shorter than 1 minute and two meals where the Shimmer3 failed to record valid data for more than half of the self-reported duration of the meal. A total of 70 of 1,133 meals (6.2%) were excluded.

#### 4.4.3 Eating Activity Classification

Of the 1,063 remaining meals, 946 triggered a positive detection (89%), and 117 meals were missed. A total of 4,966 false positives were triggered (5 false positives for every true positive). While this number seems high, it is important to note that the classes are imbalanced, as humans eat only 5% of the time. A 20:1 class imbalance causes challenges in balancing false positives and false negatives, in that if false positives occur equally with false negatives, the method could have a low false positive rate while completely missing all actual meals. For example, suppose there were only 3 false positives per day and 3 false negatives; assuming only 3 meals were consumed that day, they would all have been missed. With a high imbalance in data, accuracy must be balanced. These results are tabulated in table 4.4.

Table 4.4: Confusion matrix for the Clemson All-day data set (CAD) (hours).

		Labeled Class	
		Eating	Non-Eating
Predicted Class	Eating	190	1186
	Non-Eating	49	3256

Table 4.5: The effect of secondary activities on meal detection

Secondary Activity	Number of meals	% duration detected	% meals detected
No	544	83	91
Yes	581	78	87

While Dong’s method [56] performs with a weighted accuracy of 74% on our free-living dataset, our new method which detects secondary activities like walking and resting, improves weighted accuracy to 77%. Table 4.4 shows the confusion matrix for the new method. Weighted accuracy had a mean of 75% per subject, and a median of 78%. Approximately 72% of the subjects had weighted accuracy of 70% or higher. Paired samples t-test results show that the change in accuracy per participant due to walking and rest detection is significant ( $t[353]=7.86$ ,  $p<0.001$ ), largely due to a 23% reduction in the number of false negatives.

To evaluate the impact of secondary activities on the classification of meals, we split all meals into two categories of secondary activity: yes or no. Secondary activities included anything that could affect wrist motion for significant periods of time such as working or driving. Talking occurs frequently during free-living meals and may involve some wrist motions caused by gesturing, but we hypothesize that these wrist motions are relatively infrequent compared to those used for consumption. Therefore meals with descriptions of secondary activities that included only talking were grouped with meals with no secondary activities. Table 4.5 shows the results. Meals with no secondary activity were detected more frequently (91% vs 87%) and the total duration of eating conducted with no secondary activity was detected better than the total duration with secondary activities (83% vs 78%).

The presence of secondary activities also affects the presence of wrist activity peaks before

Table 4.6: Time difference between logged meal start and end times, and nearest peaks in minutes

Dataset	Start Time Difference	End Time Difference
Dong et al. [56]	-0.6	1.5
Free-living (this work)	-4.5	7.3

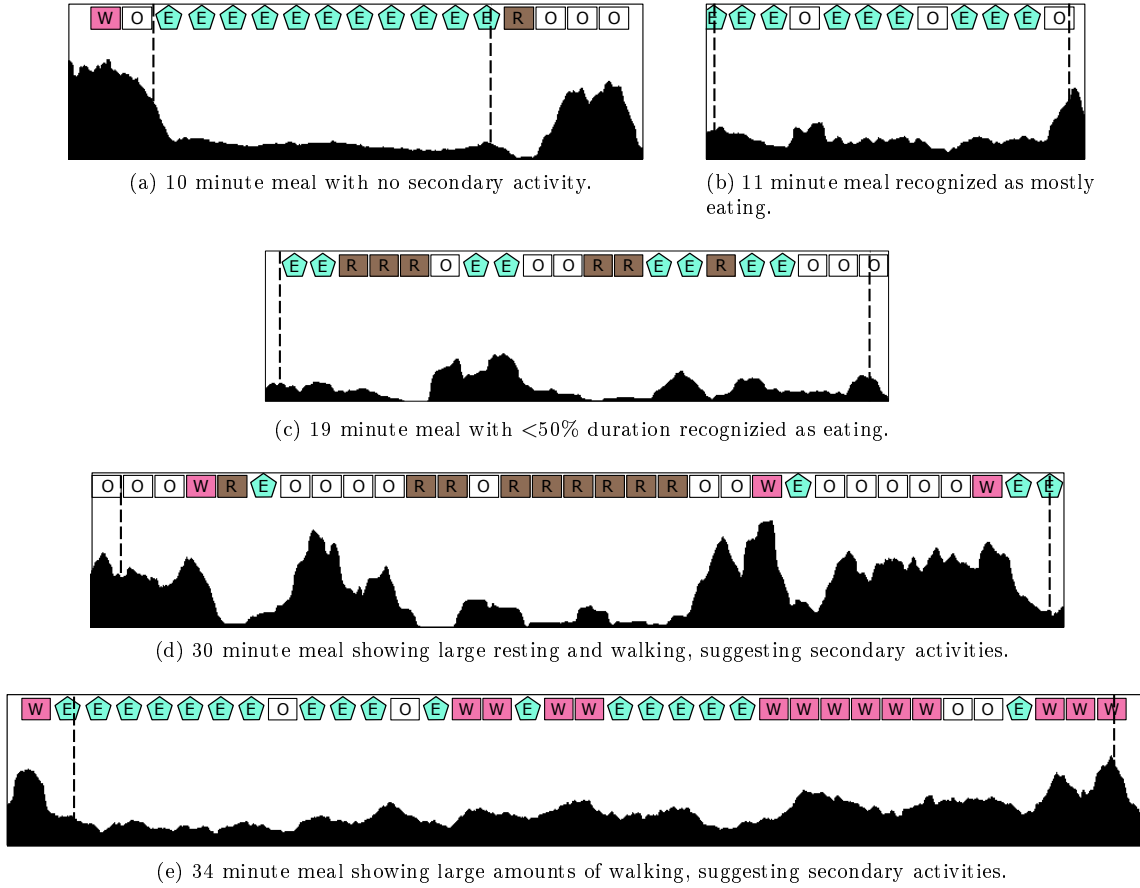


Figure 4.9: Five example meals from our free-living dataset, in increasing order of complexity. The bar on top of each figure shows majority activity detected for one minute segments (walking (W), resting (R), eating (E) or other secondary activity (O)). Meal start and end times are indicated by vertical dashed lines.

and after a meal [56]. While Dong et al. [56] reported differences of -0.6 minutes and 1.5 minutes between peaks and the start and end of self-reported meals respectively, we see differences of -4.5 and 7 minutes (table 4.6). The reason for this change is likely the presence of intermittent secondary activities during free-living meals, enabled by the low weight of the Shimmer3 (24 g), compared to the iPhone used by [56] which weighs 140 g.

#### 4.4.4 The Anatomy of a Free-living Meal

We visualized the effect of secondary activities on meals with the help of a one minute classifier. Figure 4.9a shows a 10 minute meal recognized as all eating. This is similar to what could be expected in controlled experiments where subjects consume food without conducting secondary

activities. Figure 4.9b shows an 11 minute meal in which the majority of the time is recognized as eating, but a few minutes are recognized as other. It is likely that the subject engaged in brief secondary activities. Figure 4.9c shows a 19 minute meal in which less than half the duration is recognized as eating. It is likely that the subject engaged in passive secondary activities such as watching television, as evident from the long periods of rest. Figure 4.9d shows a 30 minute meal in which only 3 minutes are recognized as eating. It is likely that the subject engaged in multiple secondary activities, including passive (as evident from the long periods of rest) and active (as evident from the long periods of other). Figure 4.9e shows a 34 minute meal in which long periods of time were recognized as walking. It is likely that the subject was multitasking, such as walking around and doing chores while simultaneously eating. It is important to note that all of these are conjecture and that we do not have minute-level ground truth of secondary activities.

## 4.5 Discussion

To our knowledge, the Clemson All-day data set (CAD) collected for this work is the largest of its kind, consisting of 4,680 hours of free-living, containing 1,133 meals/snacks, from 351 subjects. Analysis of self-reported descriptions of activities during eating indicated that 72% of all meals were consumed while performing a secondary activity (25% talking, 47% other). By detecting two common secondary activities, walking and resting, we were able to more accurately detect eating events, demonstrating the need to account for secondary activities in free-living experiments on eating detection.

Some previous works (table 4.1) have seen a drop in performance when transitioning from the lab to free-living. Our work shows that secondary activities may be contributors to this performance drop. In our work, of 408 subjects recruited, only 351 subjects (86%) provided useable data because the subjects failed to comply with instructions, or the device failed. This provides some insight into how often a wrist-worn device might be expected to fail or be used improperly.

The activities considered secondary in this chapter are specific to wrist-based sensing modalities. However, secondary activities would likely impact experiments based on other modalities. For example, methods based on sound detection at the throat [122] could consider long periods of talking a secondary activity conducted during what a person considers part of a meal. Driving and walking could affect many sensor modalities via background noise and rhythmic motions. Meals eaten while

conducting house chores or simultaneously preparing to leave the home during the morning could also affect other modalities.

As reported in table 4.6, the difference between peaks and meal boundaries was much larger in our new dataset compared to [56]. Some subjects would rest for extended periods of time after eating or worked on a computer. This results in a lack of peaks in wrist activity. We hypothesize that this segmentation impacts the classification performance of the algorithm, and plan to investigate other segmentation methods in future work.

A limitation of this work is that only coarse descriptions of secondary activities during meals were captured (e.g. ‘standing’, ‘talking to friends in a classroom’, ‘watching netflix’). Ground truth on *when* walking or resting happened during a meal was not collected due to the difficulty of collecting such information during free-living. However, our tests on the cafeteria and pedometer datasets showed very high accuracy in detecting walking and resting compared to video-based ground truth. We also specifically chose walking and resting as secondary activities due to the distinctiveness of their motion patterns. Resting can be detected by a lack of sensor motion and walking can be detected by its rhythmic motion. We believe that the combination of large datasets and readily discernible differences in the motion patterns of these activities provides confidence in the translation of our walking and resting detectors from the semi-controlled datasets to the free-living dataset. For future work, we believe it will be important to collect not only the types of secondary activities conducted during eating in free-living, but also exactly when they happen.

To conclude, by analyzing a very large free-living dataset, we learned that secondary activities during meals are common, and might not be captured in controlled or semi-controlled environments due to the lab coat effect. We show that one way to address the phenomena of secondary activities is to augment a classifier to recognize these activities. An alternative may be to model eating as a multiclass activity rather than a single class while using classifiers like a neural network or a support vector machine.

## Chapter 5

# Using deep learning to detect periods of eating

In this chapter we describe new method to automatically detect periods of eating using a convolutional neural network. A 2019 survey found that support vector machines (SVM, N=21), random forests (N=19), decision trees (N=16), and hidden Markov models (HMM, N=10) were the most commonly used classifiers in the field of detecting eating by tracking upper limb motion [123]. Neural networks (NNs) are having a tremendous impact in the fields of computer vision and healthcare [124, 125], but the 2019 survey found only 5 works that used deep learning to analyze wrist motion for detecting eating [123]. Fontana et al. [92] extracted 68 features over 30 second windows from multiple sensors (jaw motion sensor, hand proximity sensor, chest accelerometer), and then processed this information using an NN [92]. Gao et al. extracted 18 features from sound data collected by off-the-shelf bluetooth earphones to detect periods of eating [126]. Both these methods first extracted features using traditional techniques and then used an NN for classification. Kyritsis et al. developed an end-to-end deep learning approach using wrist motion data [93]. It uses two neural networks. The first network detects sub-components of hand-to-mouth gestures (raising hand, lowering hand, etc.) while the second network recognizes complete hand-to-mouth gestures (henceforward called bites). Experiments conducted in a laboratory setting found that the method could detect bites with an  $F_1$  score of 0.91 [93]. In more recent work they tested an end-to-end network in free living conditions [86]. A CNN was trained to look at 5 second windows of IMU

data and predict the probability of the window containing a bite. Hypothesizing that the density of detected bites is high during meals, and low outside periods of eating, they identify periods of time during the day as eating with 95% weighted accuracy on their own dataset [86], and 79% on a larger dataset [73].

As discussed in chapter 1.11, most previous works take a bottom-up approach to detecting eating episodes. This is sometimes called gesture spotting. Instead, we take a top-down approach by analyzing much longer windows (0.5-15 min). During a meal, we expect the window to contain gestures that are related to eating but that are not necessarily ingestion events. Examples include preparing foods for consumption, stirring, cutting, and resting between ingestion events [54, 95]. In previous work we analyzed 5 types of gestures (bite, drink, rest, utensiling, and other), and found that knowledge of the preceding gesture can improve the recognition accuracy of the following gesture using an HMM [127]. However, this work was limited to data that was manually segmented at the start and stop boundaries of gestures. In another previous work we demonstrated that a top-down approach could segment eating episodes automatically by identifying peaks of wrist motion at the beginning and end of meals caused by meal preparation and cleanup [56]. However, this work used a Bayesian classifier that averaged feature values across the entire duration of the segmented period. In this work we take a top-down approach using a CNN to automatically detect and segment periods of eating. We evaluate this idea on our new data set (CAD).

## 5.1 Methods

Figure 5.1 provides an overview of our method. We use a CNN to analyze a window of sensor data of length  $W$  min to determine the probability of eating  $p(t)$ . The window is slid  $S$  sec to generate a continuous probability of  $p(t)$  throughout the day. Figure 5.2 shows how we process this probability using hysteresis (dual thresholds) to detect periods of eating. A meal is detected if the probability of eating is higher than threshold  $T_S$  (start meal). A detected meal is ended if the probability of eating becomes lower than threshold  $T_E$  (end meal). The use of two thresholds helps reduce false positives and increase precision in the detected boundaries.

In this section we discuss the dataset used in our work. We then show how the CNN is trained to detect the probability of eating, and the hysteresis based segmenter that detects eating episodes. Finally, we discuss the metrics used to evaluate our work, and evaluate the performance



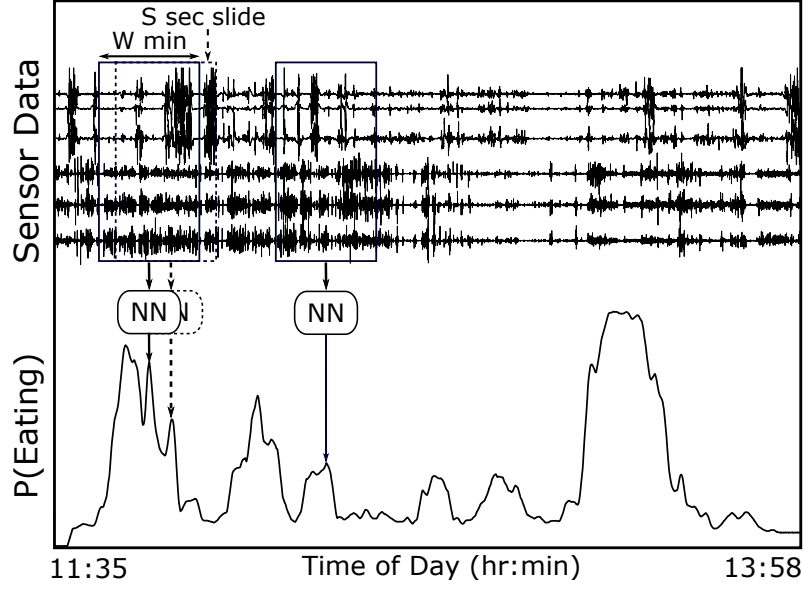


Figure 5.1: We use a sliding window of size  $W$  minutes and slide  $S$  sec to generate a continuous probability of eating for the entire day (solid line).

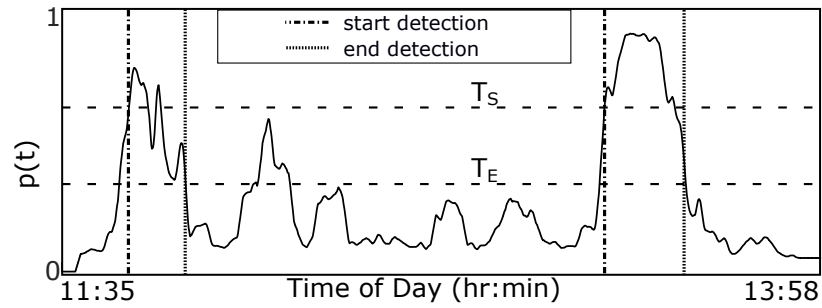


Figure 5.2: Probability of eating throughout the day is converted to periods of eating (vertical dashed lines) using a hysteresis based detector with thresholds  $T_S$  and  $T_E$ .

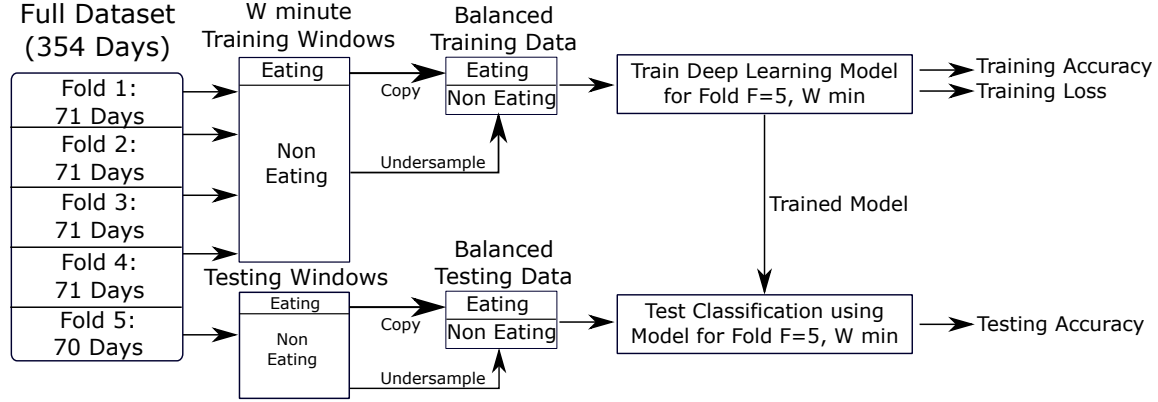


Figure 5.3: Cross-folding and data preparation for fold  $F=5$  when training the deep learning model for a window of size  $W$  minutes.

of our dataset in detecting the the 1,063 valid meals in CAD.

### 5.1.1 Pre-processing

We follow the same pre-processing as outlined in section 4.3.2. We first smooth the data on each axis individually using a Gaussian filter, using a value of  $\sigma = 10$  samples. All axis are independently normalized using means and standard deviations over the entire data set using the z-norm. Z-norm was found to yield better results than min-max scaling in related work [128].

### 5.1.2 Cross-folding and data balance

Figure 5.3 illustrates our training and testing process. The 354 days of data in CAD were split into 5 folds. Each fold ( $\approx 71$  full-day recordings) is used once for testing while the other 4 folds ( $\approx 283$  full-day recordings) are used for training. This cross-folding ensures there is no leak between the training and testing data at any step.

The CNN trained for window size  $W$  looks at a window containing  $N = W \times 60 \times 15$  samples of IMU sensor data, and predicts if the window is eating or non-eating. We use a sliding window of length  $W$  min and slide  $S_{train} = 15$  sec to generate windows for training. Each window is labeled as  $y_i = 1$  (eating) if more than 50% of the window overlaps with a self-reported meal, or  $y_i = 0$  (non-eating) otherwise. After training, we generate the probability of eating all day, using the output ( $y_i$ ) as the probability of eating at the center of the window  $p(t)$ . Since non-eating happens more often than eating, the generation of windows results in an imbalanced dataset. It is well

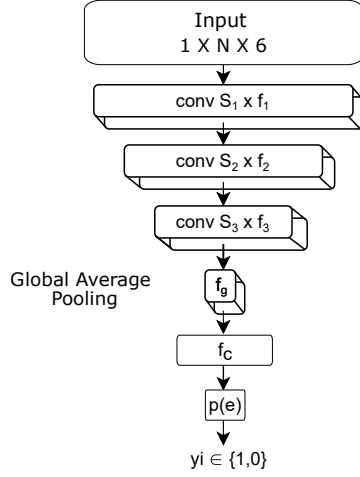


Figure 5.4: Neural network design

known that neural networks perform poorly with imbalanced datasets. We undersample (uniform sampling without replacement) the non-eating class to balance our dataset. When using 5-fold cross validation, the above data preparation step results in  $\approx 100,000$  windows for training, and  $\approx 25,000$  windows for testing in each fold, with each set containing an equal number of eating and non-eating windows.

### 5.1.3 Neural Network

We trained separate CNNs for 15 different window sizes,  $W = \{7.5 \text{ sec}, 15 \text{ sec}, 30 \text{ sec}, 45 \text{ sec}, 1 \text{ min}, 1.25 \text{ min}, 1.5 \text{ min}, 1.75 \text{ min}, 2 \text{ min}, 4 \text{ min}, 6 \text{ min}, 8 \text{ min}, 10 \text{ min}, 12 \text{ min}, 15 \text{ min}\}$ . We used the architecture described in Figure 5.4. The network has 3 convolutional layers, followed by a global pooling layer, a dense fully connected layer, and a final output layer. All layers use a relu activation [129], except for the last layer which uses a sigmoid activation [130] to output the class (eating or non-eating). All convolutional layers use a stride of 2 units and use the L1 norm for regularization [131]. The final model contains appx. 7,500 trainable parameters.

Previous work has hypothesized that lower layers of a CNN learn low level features such as edges or gradients, while deeper layers learn features such as shapes or faces. Our CNN is designed in the same spirit. Our first convolutional layer uses  $f_1 = 10$  filters, each of size  $S_1 = 44$  units equivalent to 3 sec of data at 15 Hz. This first layer is intended to learn patterns related to individual bites. Previous works found that that average bite length was 3 sec [54, 127]. The second convolutional

---

**Algorithm 1:** Pseudo-code for hysteresis based detector used to segment eating episodes.

---

**Data:** Instantaneous probability of eating during the day  $p_t$  for  $T$  data, start threshold  $T_S$ , end threshold  $T_E$   
**Result:**  $s$  detected segments of eating,  $startIndex[s]$ ,  $endIndex[s]$   
 $s \leftarrow 0$ ;  
 $t \leftarrow 0$ ;  
**while**  $t < T$  **do**  
    **while**  $p_t < T_S$  **do**  
         $t \leftarrow t + 1$ ;  
    **end**  
     $startIndex[s] \leftarrow t$ ;  
    **while**  $p_t \geq T_E$  **do**  
         $t \leftarrow t + 1$ ;  
    **end**  
     $endIndex[s] \leftarrow t$ ;  
     $s \leftarrow s + 1$ ;  
**end**

---

layer uses  $f_2 = 10$  filters of size  $S_2 = 20$ . The second layer is intended to learn patterns associated with consecutive gestures. Previous work demonstrated that knowledge of a preceding gesture can improve classification accuracy of a following gesture [127]. The third layer uses  $f_3 = 10$  filters of size  $S_3 = 4$ . This third layer is intended to learn longer patterns that may happen in entire meals. The resulting layer is pooled using global average pooling to  $f_g = 10$  numbers. Finally, the network is completed by a fully connected layer with  $f_c = 200$  units, and an output layer containing 1 unit. We use sigmoid activation on the last layer.

We did not use recurrent networks (RNNs) in our network. Our goal with network design was to find the architecture with the fewest parameters (and thus have low training and inference times), without sacrificing performance or over-fitting to the training set. RNNs increase model complexity and training time. Previous work has also shown that CNNs can perform better than RNNs in detecting long activities, while RNNs can perform better when detecting short activities like gestures [132].

We used Keras [133] to develop and train our CNN models. We use binary cross entropy loss [134], and the commonly used adam optimizer [135]. Each model was trained for 150 epochs, and the model with the best test set accuracy was retained. For hardware, we used 15 Intel Xeon E5-2680v3 computers with NVIDIA Tesla K40 GPU Computing Processors and 125 GB of RAM.

### 5.1.4 Detecting Eating Episodes

For each day of data, and for each window size  $W$ , we generated a continuous probability of eating  $p(t)$  using a sliding window of size  $S = 1$  datum. Values for  $p(t)$  range between 0 (non-eating) and 1 (eating). The value at time index  $t$  corresponds to the center datum of the sliding window. This is shown in figure 5.1.

We detect periods of eating using a hysteresis based detector. Pseudo-code for the detector is shown in algorithm 1. When  $p(t)$  goes higher than  $T_S$  (start threshold), we mark the start of a detected eating episode. When  $p(t)$  goes less than  $T_E$  (end threshold), we mark the end of an eating episode. The use of two thresholds serves two purposes. First, it helps reduce false positives by requiring a strong probability ( $> T_S$ ) for at least part of the detection. Second, it helps improve boundary precision by allowing for a weaker probability at the end of the detection. Eating tends to be more vigorous at the beginning of a meal, fading with satiety. For example, a person may slow eating with fewer ingestion events towards the end of a meal. The second threshold  $T_E$  allows for a lower probability towards the end of a detected episode. Finally, we combine any two segments that are within 1 min of each other to help avoid oversegmentation.

### 5.1.5 Evaluation Metrics

Metrics for the CNN classifier are evaluated as previously described in the introduction (section 1.10).

## 5.2 Results

We first report results while varying the parameters  $W$ ,  $T_S$  and  $T_E$ . We trained each parameter individually to identify a recommended value. We then report results of our method using the recommended values and compare to other works.

### 5.2.1 Effect of Window Size $W$

Figure 5.5 shows the effect of varying the parameter  $W$  for classifying individual windows as eating in the test folds. Large window sizes (4 - 15 min) that look for longer patterns associated with eating yield a higher classification accuracy (81-82%) than small window sizes (7.5 sec – 15 sec,

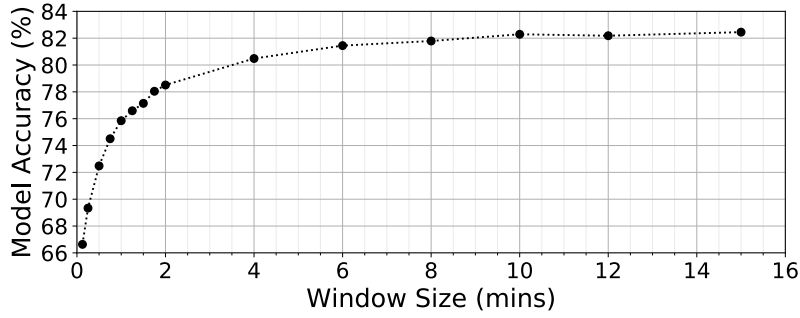


Figure 5.5: Average accuracy (5-fold) of CNN on windows from test split vs window size  $W$ .

66-69%) that look for individual ingestion events. We see that  $W > 6$  min does not further improve accuracy. However it does increase inference, prediction and training times. Thus we recommend the value of  $W = 6$  min.

### 5.2.2 Effect of Parameters $T_S$ and $T_E$

Figure 5.6 shows the effect of varying  $T_S$  on TPR and FP/TP, while  $W = 6$  min and  $T_E = 0.3$  are kept constant. As expected, lower values of  $T_S$  cause more detections. This leads to more TPs and FPs. At  $T_S=0.8$ , there is a knee in the curve where an increase in TPR is offset by an increase in FP/TP. Thus we recommend the value  $T_S=0.8$ .

Figure 5.7 shows the effect of varying  $T_E$  on average boundary error and FP/TP, while  $W = 6$  min and  $T_S = 0.8$  are kept constant. We see a general trend where low  $T_E$  values yield larger boundary errors but fewer FPs because individual detections are longer. As  $T_E$  is increased, a knee is found at  $T_E=0.4$  balancing the two metrics. Thus we recommend the value  $T_E=0.4$ .

All further results are reported using parameter values of  $W=6$  min,  $T_S=0.8$  and  $T_E=0.4$ .

### 5.2.3 Eating Episode Detection Performance

Table 5.1 shows the performance of our method when evaluated using episode level metrics. Of 1,063 total meals in the CAD data set, we detect 944 meals, miss 119, and trigger 1,650 FP. This results in a TPR of 0.89, at a 1.7 FP/TP. The only other work to analyze this data set used a Bayesian classifier and achieved the same TPR but had 5.2 FP/TP which is 67% more false positives [95]. Thus our method has achieved the best performance to date on this data set.

Comparison against methods tested on other data sets must be interpreted with caution. A

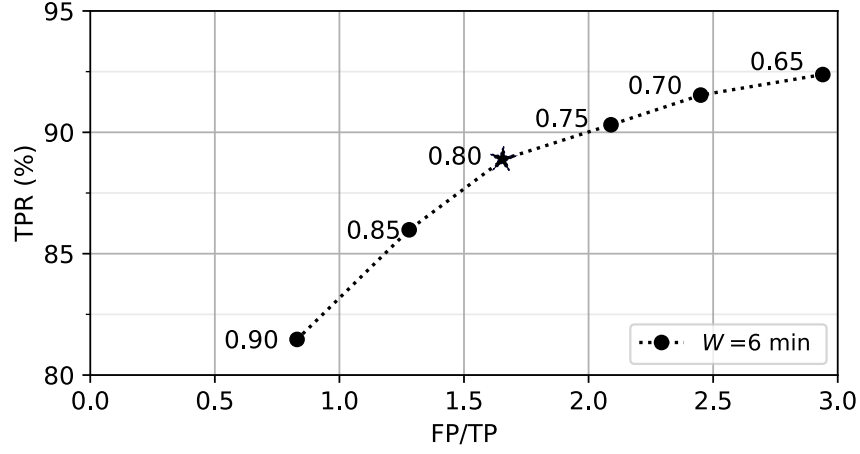


Figure 5.6: Effect of  $T_S$  on TPR and FP/TP while keeping  $W = 6$  min and  $T_E = 0.3$ . ★ indicates recommended value.

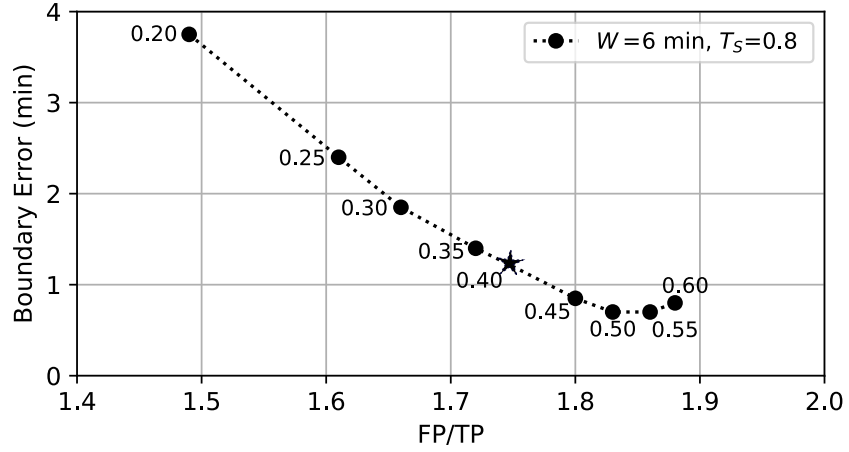


Figure 5.7: Effect of  $T_E$  on average boundary error and FP/TP while keeping  $W = 6$  mins and  $T_S = 0.8$ . ★ indicates recommended value.

Table 5.1: Eating Episode Metrics For Select Previous Work Using Wrist Tracking Data

Work	EA	Subjects	TPR (%)	FP/TP	Dataset
Thomaz 2015 [71]	-	7	-	-	Wild-7 [71]
Thomaz 2015 [71]	-	1	-	-	Wild-Long [71]
Kyritsis 2020 [86]	6	6	-	-	FreeFIC held-out [86]
Kyritsis 2020 [86]	17	6	-	-	FreeFIC [86]
Mirtchouk 2017 [73]	31	5	94	-	ACE-E/FL [73]
Mirtchouk 2017 [73]	55	6	87	-	ACE-E [73]
Kyritsis 2020 [86]	86	11	-	-	ACE-E+FL [73]
Dong 2014 [56]	116	43	86	3.8	iPhone [56]
Bayesian	1063	351	89	5.2	CAD [95]
CNN	1063	351	89	1.7	CAD [95]

Table 5.2: The CNN based classifier shows a 25% improvement in  $F_1$  score, and 6% improvement in  $ACC_W$  compared to the Bayesian classifier from Chapter 4. Results on 1,063 valid meals in CAD.

Method	Precision (%)	Recall (%)	TNR (%)	$F_1$ score (%)	$ACC_W$ (%)
Bayesian (Chapter 4)	14	76	73	23	74
CNN	36	69	93	48	80

data set containing few meals and few subjects has low within-subject and between-subject variability. This results in less variation in wrist motion patterns. Several previous works have found large performance decreases when trained on small laboratory data sets and subsequently tested in free living conditions [57, 71, 73, 118]. It is questionable if a classifier trained on a small data set would perform similarly when tested on a larger data set. Our CAD data set is much larger than all other data sets, but it has only been publicly available since 2020. We hope that more future works will use it to enable better comparisons.

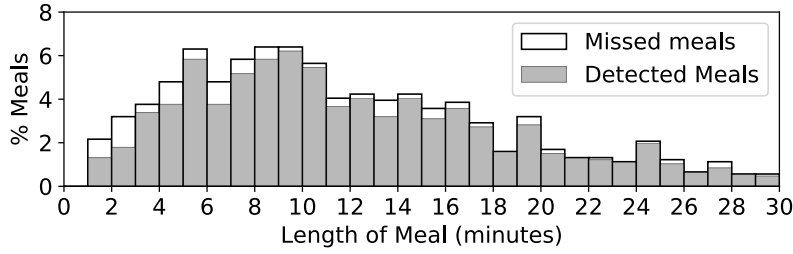
#### 5.2.4 Boundary Error Performance

The start boundary error of our method is -1.5 min, meaning that on average detected segments start 1.5 min before the self-reported start time for a meal. This is 66% reduction in error compared to previous work on CAD [95]. The end boundary error is 0.9 min, meaning that the end of detected meals came on average 0.9 min after the end of the corresponding self-reported meals. This is a 88% reduction in error compared to previous work on CAD [95].

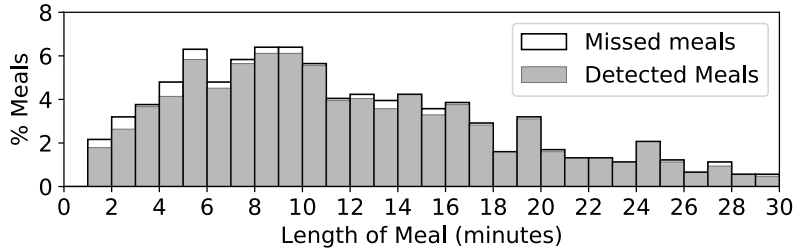
#### 5.2.5 Eating Time Detection Performance

Table 5.2 shows the performance of our method when evaluated using time metrics. Compared to the replication experiment (baseline) and the bayesian classifier (Chapter 4), our method achieves better performance on all metrics except a slight decrease in recall. We see an improvement of 22% in precision, 20% in TNR, 25% in the  $F_1$  score, and 6%  $ACC_W$ . Similar to what we found in episode level metrics, our new method produces far fewer false positives which greatly improves the time metrics.





(a) Histogram showing length of detected and missed when using  $W=6$  min,  $T_S=0.8$  and  $T_E=0.4$  (recommended parameters)



(b) Histogram showing length of detected and missed meals when using  $W=2$  min,  $T_S=0.75$  and  $T_E=0.55$

Figure 5.8: Histograms showing lengths of self-reported meals in CAD and the percent of meals detected

### 5.3 Conclusion

In this chapter we demonstrated how a top down approach to eating detection can work by creating a CNN that analyzes large windows of wrist activity data. Compared to previous attempts on CAD, we showed the best performance in eating episode metrics and time metrics.

A limitation of our work is the requirement of high computational power. Our method was trained, tested and run on a high performance computing facility, and it is reasonable to question if a real-time implementation is feasible. Kyritsis et al. have discussed this question before, and noted that a real-time implementation of their LSTM network containing 160,000 parameters may be possible on Android smartwatches if data is processed in small batches [86]. While this remains a question for future work, our proposed network contains less than 7,500 parameters and no RNN / LSTM architecture, which suggests that a real-time implementation may be possible.

The CNN proposed in chapter also has a limitation of not performing well with snacks. Using a window size of  $W=6$  min for the CNN assumes that at least 3 min of time is spent eating. Note this may include ingestion and ingestion-related gestures, but an eating episode that is shorter than this may not be detected at all. Figure 5.8(a) shows a histogram of the lengths of meals in

CAD. Of 57 meals of length less than 3 minutes, 24 (42%) are missed by our method. A trend of missing more shorter meals than larger meals can also be seen. Figure 5.8(b) shows the same histogram for our method using parameters  $W=2$  min,  $T_S=0.75$  and  $T_E=0.55$ . This reduces the number of missed snacks to just 10 (18%) and increases the episode detection rate to 95%. However, it also increases episode FP/TP to 6.1 which may not be acceptable in practical use.

Previous work has also noted how snacks are difficult to detect and can cause false positives [73]. Chapter 6 discusses issues and the lessons we learned further.

## Chapter 6

# Discussion and Conclusion

In this work we have developed a new method that can detect periods of eating by tracking wrist motion data. It takes a top-down approach by analyzing a large window (0.5-15 min) to determine if eating occurred. This can be contrasted against a bottom-up approach that analyzes a small window (1-5 sec) to find individual ingestion events, and then grouping them to identify the entire eating episode. Our top-down approach can take advantage of motions that occur *between* ingestion events such as manipulating food in preparation for ingestion and resting between ingestion events [127]. Results indicate a 15% improvement in identifying if larger windows contain eating compared to smaller windows.

Our method uses a 3 layer CNN for classification. This allows features within the large window to be modeled at different levels of granularity. This can be contrasted against a previous top-down approach that used a Bayesian classifier and modeled all features using averages across the large window [56]. On the CAD data set, our new method improves performance compared to the Bayesian classifier across episode detection metrics, boundary error, and time metrics.

Comparison between methods using different data sets should be interpreted cautiously. Data sets containing a small number of hours or participants may not contain sufficient variability in wrist motion patterns to generalize to larger testing. Our CAD data set is by far the largest of its kind, but it took us several years to collect, clean, and recently make it publicly available [95]. We hope that future work may benefit from comparisons using this data.

In Chapter 4 we showed the prevalence of secondary activities during eating, such as simultaneously walking while eating, watching media, or doing office work or chores. We showed how

Table 6.1: Precision and  $F_1$  show a downward trend as data set sizes increase. Table sorted by hours of eating data in data set used. We show selected work from the field of detecting eating using wrist motion tracking data.

Row	Work	Eating Hours	Total Hours	Precision	Recall	F1-Score	Dataset
1	Kyritsis 2020 [86]	1.6	35	86	94	90	FreeFIC held-out [86]
2	Thomaz 2015 [71]	2	32	67	89	76	Wild-7 [71]
3	Kyritsis 2020 [86]	5	77	88	92	90	FreeFIC [86]
4	Mirtchouk 2017 [73]	12	144	25	83	38	ACE-E [73]
5	Thomaz 2015 [71]	16	422	65	79	71	Wild-Long [71]
6	Kyritsis 2020 [86]	20	250	46	63	53	ACE-E+FL [73]
7	Mirtchouk 2017 [73]	20	254	31	87	46	ACE-E/FL [73]
8	Bayesian (Chapter 4)	250	4680	14	76	23	CAD
9	CNN (Chapter 5)	250	4680	36	69	48	CAD

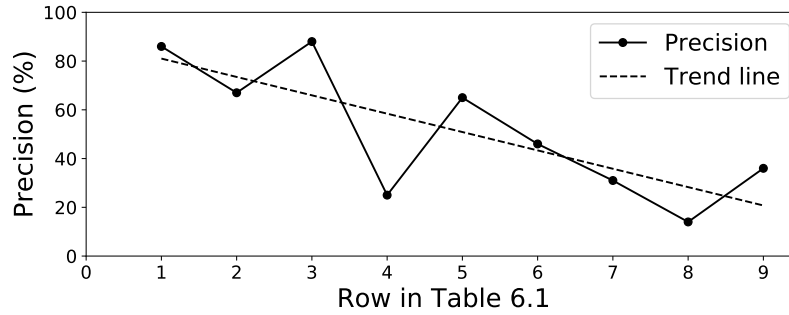


Figure 6.1: Precision reported in the literature (Table 6.1) drops as the data set sizes increase. We use row number on the X axis instead of hours of eating data as 250 hours in the last row skews the plot unreasonably.

secondary activities can obscure eating in wrist motion patterns. and that on average 12.8% of a self-reported eating episode consisted of resting and 5.5% consisted of walking. This indicates that people can spend a sizable portion of what they consider a “meal” doing things other than eating. This will have a negative impact on time metrics because those metrics evaluate every datum as eating during the self-reported period of time. In this work our new method detected 89% of eating episodes but only detected 69% of eating time (recall). We hypothesize this may in part be due to secondary activities during self-reported periods of eating.

A limitation of our work is that it may not work well on brief periods of eating such as snacks. Mirtchouk et al. reported similar issues when detecting snacks, noting that 67% of their false positives were less than 3 min in length [73]. They concluded that a different model might be needed to detect snacks. Given our insights in Chapter 5, we agree with their conclusion.

Finally, we conclude that further top-down approaches for meal detection should be explored. Non-eating (secondary) activities in the proximity of eating may contain additional information valuable for the task of detecting eating episodes (Chapter 4). We note that CAD only contains the start and end boundaries of self-reported meals. Future data collection may seek additional information such as the boundaries of meal preparation and cleanup. Modeling and classification of these activities may further improve the detection of eating episodes. We would also like to reiterate that researchers must continue to transition the field from small data sets to big data sets, as there is no evidence that methods that work on data in small data sets perform satisfactorily on data in large data sets.

# Bibliography

- [1] G. M. Turner-McGrievy, S. Wilcox, A. Boulté, B. E. Hutto, C. Singletary, E. R. Muth, and A. W. Hoover, “The dietary intervention to enhance tracking with mobile devices (diet mobile) study: A 6-month randomized weight loss trial,” *Obesity*, vol. 25, no. 8, pp. 1336–1342, 2017.
- [2] A. R. Omran, “The epidemiologic transition: a theory of the epidemiology of population change,” *The Milbank Quarterly*, vol. 83, no. 4, pp. 731–757, 2005.
- [3] Deccan Chronicle, “Apple Watch saves 76-year-old man from a heart attack,” May 2018. [Online]. Available: <https://www.deccanchronicle.com/technology/gadgets/140518/apple-watch-saves-76-year-old-man-from-a-heart-attack.html>
- [4] E. Killham, “Apple Watch knows a heart attack when it sees one,” May 2016. [Online]. Available: <https://www.cultofmac.com/417649/apple-watch-saves-man-from-heart-attack>
- [5] Z. Hall, “How Apple Watch saved one man’s life — and how it’s empowering him after his heart attack,” Dec. 2017. [Online]. Available: <https://9to5mac.com/2017/12/15/apple-watch-saves-life-managing-heart-attack>
- [6] Bloomberg, “The \$250 Biohack That’s Revolutionizing Life With Diabetes,” Nov. 2018. [Online]. Available: <http://www.bloomberg.com/news/features/2018-08-08/the-250-biohack-that-s-revolutionizing-life-with-diabetes>
- [7] C. M. Hales, M. D. Carroll, C. D. Fryar, and C. L. Ogden, *Prevalence of obesity among adults and youth: United States, 2015-2016*. US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics, 2017.
- [8] Centers for Disease Control Prevention (CDC), “Adult obesity facts,” Aug. 2018. [Online]. Available: <https://www.cdc.gov/obesity/data/adult.html>
- [9] C. J. Lavie, R. V. Milani, and H. O. Ventura, “Obesity and cardiovascular disease: risk factor, paradox, and impact of weight loss,” *Journal of the American college of cardiology*, vol. 53, no. 21, pp. 1925–1932, 2009.
- [10] Centers for Disease Control Prevention (CDC), “Adult obesity causes & consequences,” Aug. 2018. [Online]. Available: <https://www.cdc.gov/obesity/adult/causes.html>
- [11] J. O. Hill and J. C. Peters, “Environmental contributions to the obesity epidemic,” *Science*, vol. 280, no. 5368, pp. 1371–1374, 1998.
- [12] “Public domain vectors.” [Online]. Available: <https://publicdomainvectors.org/en/free-clipart/Big-salad/44522.html>
- [13] P. Kerkar, “Liposuction vs tummy tuck: Differences based on procedure, recovery, risk, cost.” [Online]. Available: <https://www.epainassist.com/differences-and-comparisons/liposuction-vs-tummy-tuck>

- [14] S. Klein, L. Fontana, V. L. Young, A. R. Coggan, C. Kilo, B. W. Patterson, and B. S. Mohammed, "Absence of an effect of liposuction on insulin action and risk factors for coronary heart disease," *New England Journal of Medicine*, vol. 350, no. 25, pp. 2549–2557, 2004.
- [15] K. D. Hall, "Predicting metabolic adaptation, body weight change, and energy intake in humans," *American Journal of Physiology-Endocrinology and Metabolism*, vol. 298, no. 3, pp. E449–E466, 2009.
- [16] M. K. Serdula, A. H. Mokdad, D. F. Williamson, D. A. Galuska, J. M. Mendlein, and G. W. Heath, "Prevalence of attempting weight loss and strategies for controlling weight," *JAMA*, vol. 282, no. 14, pp. 1353–1358, 1999.
- [17] Maastricht Instruments BV, "Room calorimeter," Oct. 2018. [Online]. Available: <http://www.maastrichtinstruments.nl/portfolio/room-calorimeter/>
- [18] R. S. Mattfeld, "Evaluation of pedometer performance across multiple gait types using video for ground truth," Ph.D. dissertation, Clemson University, 2018.
- [19] K. Mercer, L. Giangregorio, E. Schneider, P. Chilana, M. Li, and K. Grindrod, "Acceptance of commercially available wearable activity trackers among adults aged over 50 and with chronic illness: a mixed-methods evaluation," *JMIR mHealth and uHealth*, vol. 4, no. 1, p. e7, 2016.
- [20] P. Tataranni, I. Harper, S. Snitker, A. Del Parigi, B. Vozarova, J. Bunt, C. Bogardus, and E. Ravussin, "Body weight gain in free-living pima indians: effect of energy intake vs expenditure," *International Journal of Obesity*, vol. 27, no. 12, p. 1578, 2003.
- [21] 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.
- [22] K. N. Boutelle and D. S. Kirschenbaum, "Further support for consistent self-monitoring as a vital component of successful weight control," *Obesity Research*, vol. 6, no. 3, pp. 219–224, 1998.
- [23] F. B. Hu, E. Rimm, S. A. Smith-Warner, D. Feskanich, M. J. Stampfer, A. Ascherio, L. Sampson, and W. C. Willett, "Reproducibility and validity of dietary patterns assessed with a food-frequency questionnaire," *The American journal of clinical nutrition*, vol. 69, no. 2, pp. 243–249, 1999.
- [24] D. A. Schoeller, E. Ravussin, Y. Schutz, K. J. Acheson, P. Baertschi, and E. Jequier, "Energy expenditure by doubly labeled water: validation in humans and proposed calculation," *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, vol. 250, no. 5, pp. R823–R830, 1986.
- [25] D. A. Schoeller, "Measurement of energy expenditure in free-living humans by using doubly labeled water," *The Journal of nutrition*, vol. 118, no. 11, pp. 1278–1289, 1988.
- [26] J. Speakman, *Doubly labelled water: theory and practice*. Springer Science & Business Media, 1997.
- [27] R. C. Klesges, L. H. Eck, and J. W. Ray, "Who underreports dietary intake in a dietary recall? evidence from the second national health and nutrition examination survey," *Journal of consulting and clinical psychology*, vol. 63, no. 3, p. 438, 1995.

- [28] S. W. Lichtman, K. Pisarska, E. R. Berman, M. Pestone, H. Dowling, E. Offenbacher, H. Weisel, S. Heshka, D. E. Matthews, and S. B. Heymsfield, "Discrepancy between self-reported and actual caloric intake and exercise in obese subjects," *New England Journal of Medicine*, vol. 327, no. 27, pp. 1893–1898, 1992.
- [29] A. R. Kristal, U. Peters, and J. D. Potter, "Is it time to abandon the food frequency questionnaire?" 2005.
- [30] M. Ezzati, H. Martin, S. Skjold, S. V. Hoorn, and C. J. Murray, "Trends in national and state-level obesity in the usa after correction for self-report bias: analysis of health surveys," *Journal of the Royal Society of Medicine*, vol. 99, no. 5, pp. 250–257, 2006.
- [31] F. Johnson and J. Wardle, "The association between weight loss and engagement with a web-based food and exercise diary in a commercial weight loss programme: a retrospective analysis," *International Journal of Behavioral Nutrition and Physical Activity*, vol. 8, no. 1, p. 83, 2011.
- [32] M. C. Carter, V. J. Burley, C. Nykjaer, and J. E. Cade, "Adherence to a smartphone application for weight loss compared to website and paper diary: pilot randomized controlled trial," *Journal of Medical Internet Research*, vol. 15, no. 4, p. e32, 2013.
- [33] E. Helander, K. Kaipainen, I. Korhonen, and B. Wansink, "Factors related to sustained use of a free mobile app for dietary self-monitoring with photography and peer feedback: retrospective cohort study," *Journal of medical Internet research*, vol. 16, no. 4, p. e109, 2014.
- [34] G. M. Turner-McGrievy, C. G. Dunn, S. Wilcox, A. K. Boutté, B. Hutto, A. Hoover, and E. Muth, "Defining adherence to mobile dietary self-monitoring and assessing tracking over time: Tracking at least two eating occasions per day is best marker of adherence within two different mobile health randomized weight loss interventions," *Journal of the Academy of Nutrition and Dietetics*, 2019.
- [35] A. Hoover, S. Sharma, C. Martin, A. John, E. Muth, D. Thomas, and G. M. Turner-McGrievy, "Correlation between daily bite count and weight change per week," *Under Review*, 2020.
- [36] O. Amft, M. Stäger, P. Lukowicz, and G. Tröster, "Analysis of chewing sounds for dietary monitoring," in *International Conference on Ubiquitous Computing*. Springer, 2005, pp. 56–72.
- [37] O. Amft, "A wearable earpad sensor for chewing monitoring," *Sensors*, pp. 222–227, 2010.
- [38] E. Sazonov, S. Schuckers, P. Lopez-Meyer, O. Makeyev, N. Sazonova, E. L. Melanson, and M. Neuman, "Non-invasive monitoring of chewing and swallowing for objective quantification of ingestive behavior," *Physiological Measurement*, vol. 29, no. 5, p. 525, 2008.
- [39] E. S. Sazonov and J. M. Fontana, "A sensor system for automatic detection of food intake through non-invasive monitoring of chewing," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1340–1348, 2012.
- [40] J. M. Fontana and E. S. Sazonov, "A robust classification scheme for detection of food intake through non-invasive monitoring of chewing," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, 2012, pp. 4891–4894.
- [41] M. Farooq, J. M. Fontana, A. F. Boateng, M. A. Mccrory, and E. Sazonov, "A comparative study of food intake detection using artificial neural network and support vector machine," in *2013 12th International Conference on Machine Learning and Applications (ICMLA)*, vol. 1. IEEE, 2013, pp. 153–153.



- [42] S. Päßler, M. Wolff, and W.-J. Fischer, "Food intake monitoring: an acoustical approach to automated food intake activity detection and classification of consumed food," *Physiological Measurement*, vol. 33, no. 6, p. 1073, 2012.
- [43] V. Papapanagiotou, C. Diou, L. Zhou, J. van den Boer, M. Mars, and A. Delopoulos, "A novel approach for chewing detection based on a wearable ppg sensor," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 6485–6488.
- [44] R. Zhang and O. Amft, "Regular-look eyeglasses can monitor chewing," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 2016, pp. 389–392.
- [45] O. Amft and G. Tröster, "Recognition of dietary activity events using on-body sensors," *Artificial Intelligence in Medicine*, vol. 42, no. 2, pp. 121–136, 2008.
- [46] E. S. Sazonov, S. A. Schuckers, P. Lopez-Meyer, O. Makeyev, E. L. Melanson, M. R. Neuman, and J. O. Hill, "Toward objective monitoring of ingestive behavior in free-living population," *Obesity*, vol. 17, no. 10, pp. 1971–1975, 2009.
- [47] E. S. Sazonov, O. Makeyev, S. Schuckers, P. Lopez-Meyer, E. L. Melanson, and M. R. Neuman, "Automatic detection of swallowing events by acoustical means for applications of monitoring of ingestive behavior," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 3, pp. 626–633, 2010.
- [48] P. Lopez-Meyer, O. Makeyev, S. Schuckers, E. L. Melanson, M. R. Neuman, and E. Sazonov, "Detection of food intake from swallowing sequences by supervised and unsupervised methods," *Annals of Biomedical Engineering*, vol. 38, no. 8, pp. 2766–2774, 2010.
- [49] T. Olubanjo and M. Ghovanloo, "Real-time swallowing detection based on tracheal acoustics," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4384–4388.
- [50] H. Kalantarian and M. Sarrafzadeh, "Audio-based detection and evaluation of eating behavior using the smartwatch platform," *Computers in Biology and Medicine*, vol. 65, pp. 1–9, 2015.
- [51] Y. Dong, A. Hoover, and E. Muth, "A device for detecting and counting bites of food taken by a person during eating," in *2009 IEEE International Conference on Bioinformatics and Biomedicine*. IEEE, 2009, pp. 265–268.
- [52] Y. Dong, A. Hoover, J. Scisco, and E. Muth, "A new method for measuring meal intake in humans via automated wrist motion tracking," *Applied Psychophysiology and Biofeedback*, vol. 37, no. 3, pp. 205–215, 2012.
- [53] R. I. Ramos-Garcia, E. R. Muth, J. N. Gowdy, and A. W. Hoover, "Improving the recognition of eating gestures using intergesture sequential dependencies," *IEEE journal of biomedical and health informatics*, vol. 19, no. 3, pp. 825–831, 2014.
- [54] Y. Shen, J. Salley, E. Muth, and A. Hoover, "Assessing the accuracy of a wrist motion tracking method for counting bites across demographic and food variables," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 599–606, 2017.
- [55] Y. Dong, A. Hoover, J. Scisco, and E. Muth, "Detecting eating using a wrist mounted device during normal daily activities," in *Proceedings of the International Conference on Embedded Systems, Cyber-physical Systems, and Applications (ESCS)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2011, p. 1.

- [56] Y. Dong, J. Scisco, M. Wilson, E. Muth, and A. Hoover, "Detecting periods of eating during free-living by tracking wrist motion," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1253–1260, 2014.
- [57] K. S. Chun, S. Bhattacharya, and E. Thomaz, "Detecting eating episodes by tracking jaw-bone movements with a non-contact wearable sensor," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, p. 4, 2018.
- [58] R. Zhang and O. Amft, "Retrieval and timing performance of chewing-based eating event detection in wearable sensors," *Sensors*, vol. 20, no. 2, p. 557, 2020.
- [59] TDK, "Mpu-6050." [Online]. Available: <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
- [60] —, "Mpu-9150." [Online]. Available: <https://www.invensense.com/products/motion-tracking/9-axis/mpu-9150/>
- [61] Carolyn Mathas, Digikey, "Sensor fusion: The basics," 2019, [Online; accessed 8-July-2019]. [Online]. Available: <https://www.digikey.ca/en/articles/techzone/2012/apr/sensor-fusion-the-basics>
- [62] Shimmer, "Imu user guide." [Online]. Available: [http://www.shimmersensing.com/images/uploads/docs/IMU\\_User\\_Guide\\_rev1.4.pdf](http://www.shimmersensing.com/images/uploads/docs/IMU_User_Guide_rev1.4.pdf)
- [63] G. X. Jay Esfandyari, Roberto De Nuccio, "Introduction to mems gyroscopes," STMicroelectronics. [Online]. Available: <https://electroiq.com/2010/11/introduction-to-mems-gyroscopes/>
- [64] uavnavigation.com, "Introduction to magnetometers," 2019, [Online; accessed 8-July-2019]. [Online]. Available: <https://www.uavnavigation.com/support/kb/peripherals/magnetometers/introduction-magnetometers>
- [65] I. Kononenko, "Automatic knowledge acquisition," *Current trends in knowledge acquisition*, vol. 8, p. 190, 1990.
- [66] A. Lagandula, "Rule based classifiers." [Online]. Available: <https://towardsdatascience.com/perceptron-the-artificial-neuron-4d8c70d5cc8d>
- [67] M. Minsky and S. A. Papert, *Perceptrons: An introduction to computational geometry*. MIT Press, 2017.
- [68] S. El Hihi and Y. Bengio, "Hierarchical recurrent neural networks for long-term dependencies," in *Advances in Neural Information Processing Systems*, 1996, pp. 493–499.
- [69] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [70] United States Department of Labor, "Bureau of labor statistics data for activity: Eating and drinking," 2016. [Online]. Available: <https://data.bls.gov/timeseries/TUU10101AA01013237>
- [71] 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*. ACM, 2015, pp. 1029–1040.
- [72] 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.

- [73] M. Mirtchouk, D. Lustig, A. Smith, I. Ching, M. Zheng, and S. Kleinberg, “Recognizing eating from body-worn sensors: Combining free-living and laboratory data,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. 85, 2017.
- [74] R. Zhang and O. Amft, “Free-living eating event spotting using emg-monitoring eyeglasses,” in *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 2018, pp. 128–132.
- [75] M. Farooq, A. Doulah, J. Parton, M. A. McCrory, J. A. Higgins, and E. Sazonov, “Validation of sensor-based food intake detection by multicamera video observation in an unconstrained environment,” *Nutrients*, vol. 11, no. 3, p. 609, 2019.
- [76] L. K. Heilbronn, L. de Jonge, M. I. Frisard, J. P. DeLany, D. E. Larson-Meyer, J. Rood, T. Nguyen, C. K. Martin, J. Volaufova, M. M. Most *et al.*, “Effect of 6-month calorie restriction on biomarkers of longevity, metabolic adaptation, and oxidative stress in overweight individuals: a randomized controlled trial,” *JAMA*, vol. 295, no. 13, pp. 1539–1548, 2006.
- [77] T. S. Church, C. K. Martin, A. M. Thompson, C. P. Earnest, C. R. Mikus, and S. N. Blair, “Changes in weight, waist circumference and compensatory responses with different doses of exercise among sedentary, overweight postmenopausal women,” *PloS One*, vol. 4, no. 2, p. e4515, 2009.
- [78] K. S. Button, J. P. Ioannidis, C. Mokrysz, B. A. Nosek, J. Flint, E. S. Robinson, and M. R. Munafò, “Power failure: why small sample size undermines the reliability of neuroscience,” *Nature Reviews Neuroscience*, vol. 14, no. 5, p. 365, 2013.
- [79] C. Carberry, “How to eat like a skinny person.” [Online]. Available: <https://www.wikihow.com/Eat-Like-a-Skinny-Person>
- [80] J. Windt, “The myth of multitasking.” [Online]. Available: <https://www.janicewindt.com/2018/08/27/myth-of-multitasking/>
- [81] A. Doulah, X. Yang, J. Parton, J. A. Higgins, M. A. McCrory, and E. Sazonov, “The importance of field experiments in testing of sensors for dietary assessment and eating behavior monitoring,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 5759–5762.
- [82] R. Alharbi, T. Stump, N. Vafaie, A. Pfammatter, B. Spring, and N. Alshurafa, “I can’t be myself: Effects of wearable cameras on the capture of authentic behavior in the wild,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, p. 90, 2018.
- [83] E. Thomaz, A. Parnami, J. Bidwell, I. Essa, and G. D. Abowd, “Technological approaches for addressing privacy concerns when recognizing eating behaviors with wearable cameras,” in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2013, pp. 739–748.
- [84] C. A. Merck, C. Maher, M. Mirtchouk, M. Zheng, Y. Huang, and S. Kleinberg, “Multimodality sensing for eating recognition,” in *PervasiveHealth*, 2016, pp. 130–137.
- [85] B. M. Bell, R. Alam, N. Alshurafa, E. Thomaz, A. S. Mondol, K. de la Haye, J. A. Stankovic, J. Lach, and D. Spruijt-Metz, “Automatic, wearable-based, in-field eating detection approaches for public health research: a scoping review,” *NPJ Digital Medicine*, vol. 3, no. 1, pp. 1–14, 2020.

- [86] K. Kyritsis, C. Diou, and A. Delopoulos, "A data driven end-to-end approach for in-the-wild monitoring of eating behavior using smartwatches," *IEEE Journal of Biomedical and Health Informatics*, 2020.
- [87] S. Breznitz, *Cry wolf: The psychology of false alarms*. Psychology Press, 2013.
- [88] Y. Xiao, F. J. Seagull, F. Nieves-Khouw, N. Barczak, and S. Perkins, "Organizational-historical analysis of the " failure to respond to alarm" problems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 34, no. 6, pp. 772–778, 2004.
- [89] Wikipedia contributors, "Confusion matrix - table of confusion," 2019, [Online; accessed 6-August-2019]. [Online]. Available: [https://en.wikipedia.org/wiki/Confusion\\_matrix#Table\\_of\\_confusion](https://en.wikipedia.org/wiki/Confusion_matrix#Table_of_confusion)
- [90] —, "Precision and recall - imbalanced data," 2019, [Online; accessed 6-August-2019]. [Online]. Available: [https://en.wikipedia.org/wiki/Precision\\_and\\_recall#Imbalanced\\_Data](https://en.wikipedia.org/wiki/Precision_and_recall#Imbalanced_Data)
- [91] G. Schiboni and O. Amft, "Sparse natural gesture spotting in free living to monitor drinking with wrist-worn inertial sensors," in *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. ACM, 2018, pp. 140–147.
- [92] J. M. Fontana, M. Farooq, and E. Sazonov, "Automatic ingestion monitor: a novel wearable device for monitoring of ingestive behavior," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 6, pp. 1772–1779, 2014.
- [93] K. Kyritsis, C. Diou, and A. Delopoulos, "Modeling wrist micromovements to measure in-meal eating behavior from inertial sensor data," *IEEE Journal of Biomedical and Health Informatics*, 2019.
- [94] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognition*, vol. 41, no. 6, pp. 2010–2024, 2008.
- [95] S. Sharma, P. Jasper, E. Muth, and A. Hoover, "The impact of walking and resting on wrist motion for automated detection of meals," *ACM Transactions on Computing for Healthcare*, 2020.
- [96] S. P. Sharma, "A device to record natural daily wrist motion," Ph.D. dissertation, Clemson University, 2014.
- [97] J. Overduin, "Einstein's spacetime." [Online]. Available: <https://einstein.stanford.edu/SPACETIME/spacetime2.html>
- [98] R. Mahony, T. Hamel, and J.-M. Pfimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.
- [99] V. T. Van Hees, L. Gorzelniak, E. C. D. Leon, M. Eder, M. Pias, S. Taherian, U. Ekelund, F. Renström, P. W. Franks, A. Horsch *et al.*, "Separating movement and gravity components in an acceleration signal and implications for the assessment of human daily physical activity," *PloS One*, vol. 8, no. 4, p. e61691, 2013.
- [100] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*. IEEE, 2011, pp. 1–7.

- [101] Wikipedia contributor Auawise, “Rotation formalisms in three dimensions — Wikipedia, the free encyclopedia,” 2019, [Accessed 8-July-2019]. [Online]. Available: [https://en.wikipedia.org/wiki/Aircraft\\_principal\\_axes](https://en.wikipedia.org/wiki/Aircraft_principal_axes)
- [102] P. Kelly, “Solid mechanics part iii: foundations of continuum solid mechanics,” *Lecture Notes*, 2013.
- [103] Wikipedia contributor Schorschi2, “Right-hand rule — Wikipedia, the free encyclopedia,” 2019, [Accessed 8-July-2019]. [Online]. Available: [https://en.wikipedia.org/wiki/Right-hand\\_rule](https://en.wikipedia.org/wiki/Right-hand_rule)
- [104] S. Sharma and A. Hoover, “A study on linear acceleration of the wrist during free-living,” in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019, pp. 1624–1628.
- [105] Google Tech Talks. Sensor Fusion on Android Devices: A Revolution in Motion Processing. [Online]. Available: <https://www.youtube.com/watch?v=C7JQ7Rpwn2k>
- [106] TDK Invensense. (2018) MPU 9250. [Online]. Available: <https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250>
- [107] J. A. Rios and E. White, “Fusion filter algorithm enhancements for a MEMS GPS/IMU,” *Crossbow Technology, Inc*, pp. 1–12, 2002.
- [108] P. Björkholm, L. Landén, J. Stigwall, P. Johannisson, S. Södermalm, P. Andersson, and C. Johnsson, “Navigation in vehicle crash test using MEMS-based IMU,” in *2010 IEEE/ION Position Location and Navigation Symposium (PLANS)*. IEEE, 2010, pp. 27–31.
- [109] A. R. J. Ruiz, F. S. Granja, J. C. P. Honorato, and J. I. G. Rosas, “Pedestrian indoor navigation by aiding a foot-mounted IMU with RFID signal strength measurements,” in *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2010, pp. 1–7.
- [110] A. Rowlands and V. Stiles, “Accelerometer counts and raw acceleration output in relation to mechanical loading,” *Journal of Biomechanics*, vol. 45, no. 3, pp. 448–454, 2012.
- [111] L. Ye, S. W. Su, D. Lei, and H. T. Nguyen, “An online recursive autocalibration of triaxial accelerometer,” in *Proc. of 2016 IEEE 38th Annual International Conference of the Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 2038–2041.
- [112] Invensense. (2010) Sensor fusion on android devices: A revolution in motion processing. [Online]. Available: <https://www.youtube.com/watch?v=C7JQ7Rpwn2k>
- [113] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones.” in *Esann*, 2013.
- [114] Apple Inc. Core Motion Apple Developer Documentation. [Online]. Available: <https://developer.apple.com/documentation/coremotion>
- [115] Invensense Inc. Motion sensors introduction. [Online]. Available: <https://store.invensense.com/datasheets/invensense/Sensor-Introduction.pdf>
- [116] I. Frosio, F. Pedersini, and N. A. Borghese, “Autocalibration of MEMS accelerometers,” *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 2034–2041, 2009.
- [117] C. Hirt, S. Claessens, T. Fecher, M. Kuhn, R. Pail, and M. Rexer, “New ultrahigh-resolution picture of earth’s gravity field,” *Geophysical Research Letters*, vol. 40, no. 16, pp. 4279–4283, 2013.

- [118] S. Zhang, R. Alharbi, M. Nicholson, and N. Alshurafa, "When generalized eating detection machine learning models fail in the field," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 2017, pp. 613–622.
- [119] R. Mattfeld, E. Jesch, and A. Hoover, "A new dataset for evaluating pedometer performance," in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2017, pp. 865–869.
- [120] G. Deng and L. Cahill, "An adaptive gaussian filter for noise reduction and edge detection," in *1993 IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference*. IEEE, 1993, pp. 1615–1619.
- [121] J. Sim and C. C. Wright, "The kappa statistic in reliability studies: use, interpretation, and sample size requirements," *Physical therapy*, vol. 85, no. 3, pp. 257–268, 2005.
- [122] O. Amft, M. Kusserow, and G. Troster, "Automatic identification of temporal sequences in chewing sounds," in *IEEE International Conference on Bioinformatics and Biomedicine, 2007. BIBM 2007*. IEEE, 2007, pp. 194–201.
- [123] H. Heydarian, M. Adam, T. Burrows, C. Collins, and M. E. Rollo, "Assessing eating behaviour using upper limb mounted motion sensors: A systematic review," *Nutrients*, vol. 11, no. 5, p. 1168, 2019.
- [124] G. Wang, "A perspective on deep imaging," *IEEE Access*, vol. 4, pp. 8914–8924, 2016.
- [125] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24–29, 2019.
- [126] Y. Gao, N. Zhang, H. Wang, X. Ding, X. Ye, G. Chen, and Y. Cao, "ihear food: eating detection using commodity bluetooth headsets," in *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 2016, pp. 163–172.
- [127] R. I. Ramos-Garcia, E. R. Muth, J. N. Gowdy, and A. W. Hoover, "Improving the recognition of eating gestures using intergesture sequential dependencies," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 825–831, 2015.
- [128] M. Farooq and E. Sazonov, "Detection of chewing from piezoelectric film sensor signals using ensemble classifiers," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 4929–4932.
- [129] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.
- [130] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of back-propagation learning," in *International Workshop on Artificial Neural Networks*. Springer, 1995, pp. 195–201.
- [131] M. Y. Park and T. Hastie, "L1-regularization path algorithm for generalized linear models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 659–677, 2007.
- [132] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 1533–1540.

- [133] F. Chollet *et al.*, “keras,” 2015.
- [134] M. A. Nielsen, *Neural networks and deep learning*. Determination press San Francisco, CA, USA:, 2015, vol. 2018.
- [135] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.