Dartmouth College
Dartmouth Digital Commons

Dartmouth College Ph.D Dissertations | Theses, Dissertations, and Graduate Essays

5-3-2021

DETECTION OF HEALTH-RELATED BEHAVIOURS USING HEAD-MOUNTED DEVICES

Shengjie Bi
Dartmouth College, shengjie.bi.gr@dartmouth.edu

Follow this and additional works at: https://digitalcommons.dartmouth.edu/dissertations

Part of the Other Computer Sciences Commons

Recommended Citation

Bi, Shengjie, "DETECTION OF HEALTH-RELATED BEHAVIOURS USING HEAD-MOUNTED DEVICES" (2021). *Dartmouth College Ph.D Dissertations*. 75.
https://digitalcommons.dartmouth.edu/dissertations/75

This Thesis (Ph.D.) is brought to you for free and open access by the Theses, Dissertations, and Graduate Essays at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Ph.D Dissertations by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.

DETECTION OF HEALTH-RELATED BEHAVIOURS USING HEAD-MOUNTED DEVICES

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of
Doctor of Philosophy
in
Computer Science
by
Shengjie Bi
Guarini School of Graduate and Advanced Studies
Dartmouth college
Hanover, New Hampshire
May 2021

Examining Committee:

(chair) David Kotz, Ph.D.

V.S. Subrahmanian, Ph.D.

Xia Zhou, Ph.D.

Nabil Alshurafa, Ph.D.

F. Jon Kull, Ph.D.
Dean of the Guarini School of Graduate and Advanced Studies

Abstract

The detection of health-related behaviors is the basis of many mobile-sensing applications for healthcare and can trigger other inquiries or interventions. Wearable sensors have been widely used for mobile sensing due to their ever-decreasing cost, ease of deployment, and ability to provide continuous monitoring. In this dissertation, we develop a generalizable approach to sensing eating-related behavior.

First, we developed *Auracle*, a wearable earpiece that can automatically detect eating episodes. Using an off-the-shelf contact microphone placed behind the ear, Auracle captures the sound of a person chewing as it passes through the head. This audio data is then processed by a custom circuit board. We collected data with 14 participants for 32 hours in free-living conditions and achieved accuracy exceeding 92.8% and F1 score exceeding 77.5% for eating detection with 1-minute resolution.

Second, we adapted Auracle for measuring children's eating behavior, and improved the accuracy and robustness of the eating-activity detection algorithms. We used this improved prototype in a laboratory study with a sample of 10 children for 60 total sessions and collected 22.3 hours of data in both meal and snack scenarios. Overall, we achieved 95.5% accuracy and 95.7% F1 score for eating detection with 1-minute resolution.

Third, we developed a computer-vision approach for eating detection in free-living scenarios. Using a miniature head-mounted camera, we collected data with 10 participants

for about 55 hours. The camera was fixed under the brim of a cap, pointing to the mouth of the wearer and continuously recording video (but not audio) throughout their normal daily activity. We evaluated performance for eating detection using four different Convolutional Neural Network (CNN) models. The best model achieved 90.9% accuracy and 78.7% F1 score for eating detection with 1-minute resolution. Finally, we validated the feasibility of deploying the 3D CNN model in wearable or mobile platforms when considering computation, memory, and power constraints.

Contents

List of Figures

List of Tables

List of Abbreviations

List of abbreviations used in this dissertation, with page numbers where the abbreviation is first used.

ADM Automatic Dietary Monitoring 6

AFE Analog Front End 27

AUC Area Under the Curve 71

BLE Bluetooth Low Energy 25

CNN Convolutional Neural Network iii

CV Computer Vision 79

EMG electromyography 4

FPS frames per second 81

GFLOPs 1×10^9 floating point operations per second 90

GPU graphics processing unit 84

IRB Institutional Review Board 13

LOSO Leave-One-Session-Out 65

RMSE root mean square energy 67

1

Introduction

Chronic disease is one of the most pressing health challenges faced in the United States, and around the world. According to one report, nearly half (approximately 45%, or 133 million) of all Americans suffer from at least one chronic disease, and the number is growing [54]. Chronic diseases are a tremendous burden to the individuals, their families, and to society. By 2023, diabetes alone is estimated to cost $430 billion to the US economy [13]. Many chronic diseases are an outcome of, or exacerbated by, an individual's lifestyle. Behaviors such as eating, drinking and smoking are strongly related to chronic diseases like obesity, hypertension, diabetes, lung cancer, heart disease and metabolic disorders. Scientists are still trying to fully understand the complex mixture of diet, exercise, genetics, sociocultural

context, and physical environment that lead to these diseases.

Mobile, wearable, and embedded sensing technology present an opportunity to measure health-related behaviors and help with behavior change [50]. Many commercially available wearable devices can monitor fitness activities such as walking, running, and swimming. There is, however, no commercially available device that can automatically monitor health-related activities such as eating, drinking, or smoking. The availability of such a device would be a huge benefit to health-science research.

Although researchers have proposed various approaches to monitor different health-related behaviors, it is not yet possible to accurately, automatically and seamlessly recognize many important health-related behaviors outside the lab. Thus our interest is to develop a wearable system that is effective and robust enough to automatically detect eating-related behaviors in out-of-lab, day-long, free-living conditions.

Everyone eats – and these behaviors are critical to many aspects of personal health – yet science has only a limited understanding of eating. In contrast to many commercial sensing devices that measure physical activity (*caloric output*), such as Fitbit, similar devices to track eating (*caloric intake*) have lagged behind. An ideal embodiment of a system for monitoring eating has several challenges: (a) identifying *when* and for *how long* an individual ate, (b) identifying nutritional information (e.g., *what* and *how much* food was consumed), (c) identifying further eating parameters (e.g., the number of *mouthfuls* or the *chewing rate*), and (d) ensuring that the system is *usable* in real-world settings (i.e., it is unobtrusive, energy-efficient, robust to environmental noise, and easy to use). In this proposal, we focus on unobtrusive, automatic methods for accurately identifying *when* and for *how long* an individual performs *eating* – the foundation for triggering further sensing operations or for inquiries to the user. (For instance, a wearable camera could be triggered when the eating recognition system detects eating; a digital food journal, which includes times and durations of eating and pictures of food, can be generated and sent to nutritionists for analysis.)

Indeed, we designed our system for use primarily by health-science researchers. For

instance, a health-science researcher may want to study the eating habits of college students throughout a semester: when and how often do they eat? for how long? how do these patterns change during exam periods? Our system could be used for such research purposes, and has the potential to be adjusted for detecting other health-related behaviors in similar conditions.

First, we developed an approach for eating detection in laboratory conditions. We then explored the generalization of our work along two dimensions: from laboratory conditions to free-living conditions, and from adults to children. Lastly, we developed a computer-vision based approach for eating detection in free-living conditions and explored the use of deep-learning models (CNN) rather than traditional statistical machine-learning models for eating detection.

1.1 Research questions

In this section, we first state our main research questions. We then refine each question into specific sub-questions, and identify each of our corresponding contributions. We have four main research questions:

- How to detect eating in laboratory conditions?
- How to detect eating in free-living conditions?
- How to adapt for children an eating-detection approach designed for adults?
- How to detect eating in free-living conditions using computer vision?

1.1.1 Eating detection in laboratory conditions

To develop a wearable system for eating detection in laboratory conditions, we addressed five research questions:

- What sensor (or combination of sensors) work best?
- On what body location are the sensor(s) best placed?

- What are the best values for parameters like sample rate, window size, bit resolution, number of features?
- How well does our method generalize to different types of food?
- How well does our method work for a new person not included in the training data?

To answer these questions, we compared two popular methods for eating detection (based on acoustic and electromyography (EMG) sensors) individually and combined. We built a data-acquisition system using two off-the-shelf sensors and conducted a study with 20 participants. More details can be found in Chapter 2.

This work made two main contributions:

- We compared two sensing modalities (acoustics and EMG) in terms of performance and usability for free-living scenarios. We concluded the best approach is to use the acoustic sensor, alone, because its accuracy was nearly as good as the two-sensor approach, and the EMG sensor was uncomfortable to wear and difficult to attach.
- We demonstrated the potential for implementing this system as a robust wearable for long-term use in free-living scenarios.

1.1.2 Eating detection in free-living conditions

To further improve our wearable system for eating detection in out-of-lab, day-long, free-living scenarios, we formulated the following research questions:

- What hardware components are needed?
- How to ensure reliable contact between microphone and skin?
- How to ensure comfort for persons with different head shapes?
- How to obtain "ground-truth" in the field?
- Can we run feature extraction and classification on-board, in real time?
- Can we use additional in-laboratory eating data to improve the classification results for field studies?
- How to aggregate eating-detection results of short time windows to eating episodes?

- How to improve the power efficiency of our system?

To answer the above questions, we developed ***Auracle***,[1] a wearable earpiece that can automatically recognize eating behavior. More specifically, in free-living conditions, we can recognize when and for how long a person is eating. Using an off-the-shelf contact microphone placed behind the ear, *Auracle* captures the sound of a person chewing as it passes through the bone and tissue of the head. This audio data is then processed by a custom analog/digital circuit board. To ensure reliable (yet comfortable) contact between microphone and skin, all hardware components are incorporated into a 3D-printed behind-the-head framework. Please refer to Chapter 3 for more details.

This work made three main contributions:

- We developed *Auracle*, which is the first system that demonstrates the possibility of using a self-contained, ear-mounted system with an in-built contact microphone for eating detection in free-living conditions.
- *Auracle* runs feature extraction and classification algorithms in an ultra-low-power microcontroller (MCU) (ARM Cortex M3). Previous researchers run their models for eating detection using platforms that are significantly more power hungry (such as a laptop, smartphone, or Arduino). Based on our power measurements, we estimated *Auracle* could last for 28.1 hours with a 110 mAh battery, all while transmitting eating notifications to a subject's smartphone.
- We demonstrated the success of *Auracle* in a field deployment involving 14 participants, despite challenges with environmental noise (ambient sound, motion artifacts), in a setting different from training conditions (e.g., subjects eating while walking), and with widely varying food types.

[1] http://auracle-project.org

1.1.3 Adaption of eating-detection approach for children

To adapt our approach for measuring children's eating behavior, we explored three research questions:

- What form factor can easily adapt to a range of children and ensure proper fit and comfort?
- How well does our method generalize to children in both meal and snack scenarios?
- What refinements are needed to our data-analysis approach, to achieve better performance?

To explore these questions, we identified and addressed several challenges pertaining to monitoring eating behavior in children, paying particular attention to device fit and comfort. We also improved the accuracy and robustness of the eating-activity detection algorithms. We used this improved prototype in a lab study with a sample of 10 children for 60 total sessions and collected 22.3 hours of data in both meal and snack scenarios. Please see Chapter 4 for more details.

We made two important contributions:

- We demonstrated that it is feasible to monitor the eating activity of children automatically. This result provided the foundation for behavioral researchers, clinicians, and dietitians to understand fine-grained details about a child's eating habits.
- We identified unique challenges pertaining to the use of existing Automatic Dietary Monitoring (ADM) systems (designed for an adult population) on children. We detailed the steps necessary to adapt an adult device to allow data collection from children. With these adaptations, we developed the first ADM system focused on the study of eating behavior in children.

1.1.4 Computer-vision based approach for eating detection

Lastly, we developed computer-vision based approaches to detect eating in free-living scenarios. In this work, we explored the following research questions:

- Can we detect eating in free-living scenarios from raw video frames?
- Which type of information (spatial or temporal information) is more important for CNN models to detecting eating?
- Is it sufficient to use only optical flow features to achieve good eating-detection performance?
- What CNN architecture gives the best accuracy?
- Is it possible to deploy the CNN model for eating detection in a wearable or mobile platform?

To explore these questions, we conducted a field study and collected data with 10 participants for about 55 hours. We designed a data-processing pipeline and evaluated performance of eating detection with four different CNN models. The best model achieved accuracy 90.9% and F1 score 78.7% for eating detection with a 1-minute resolution. We also discussed the resources needed to deploy a 3D CNN model in wearable or mobile platforms, in terms of computation, memory, and power. Please see Chapter 5 for more details.

This work made the following contributions:

- We developed the first video-based approach for eating detection in free-living conditions and demonstrated the success of our approach in a field deployment involving 10 participants.
- We demonstrated the feasibility of using CNN models to detect eating from raw video frames of a face viewed from an oblique angle.
- We showed that temporal context is crucial and considerably improved the performance for eating detection when using CNN models.

1.2 Note on collaboration

Our work is the outcome of collaborations and teamwork of researchers from different fields, including psychologists, health scientists, computer scientists, electrical engineers, and mechanical designers. The contributions made by all the collaborators have led to the success of the various research projects. For the purpose of this dissertation, I understand it is necessary to distinguish *my* contributions from the contributions of other team members in the project. I make the necessary distinction and list a breakdown of *my* contributions and contributions of others in the following chapters.

2

Eating detection in laboratory conditions

In this chapter, I describe my contributions to the development of a wearable method for the detection of eating, and its evaluation in a laboratory setting:

- I developed a data acquisition system and evaluated two sensors for a behind-the-ear device: a contact microphone and an EMG sensor.
- I conducted a laboratory study with 20 participants and implemented a data-analysis approach that involved multiple stages including feature extraction, feature selection and classification.
- I experimented and identified the best values for analysis parameters, including sample rate, window size, bit resolution, and number of features.

- I demonstrated that our method can detect eating with an accuracy exceeding 90.9% even when the 'crunchiness' of food varies.

Regarding the work described in this chapter, I acknowledge the contributions of others:

- Tao Wang assisted the data collection and implementation of the data-analysis approach.
- Ellen Davenport developed the headband used in the wearable apparatus.

For these above-mentioned contributions, I provided collaborative input.

2.1 Background

In most (if not all) previous reports of eating-detection technologies, researchers do not provide a precise definition of eating, even though they set out to detect eating. We define *eating* in this and following chapters as "an activity involving the chewing of food that is eventually swallowed." This definition may exclude drinking, which usually does not involve chewing. On the other hand, the consumption of "liquid foods" that contain solid content (like vegetable soup), which requires chewing, is considered eating. Our definition also excludes chewing gum, since gum is not usually swallowed.

Our goal is to develop a wearable device that can last a waking day and recognize eating in free-living scenarios. Researchers have explored several body locations for eating detection, which include inside the ear canal [1, 7, 48], against the throat [30, 55], and on the wrist [66]. To ensure user comfort for long periods of time and not impede hearing during daily activities, placing sensors inside the ear canal may not be acceptable. The throat is physically close to the location of swallowing, but placing sensors against the throat may be considered too obtrusive by users. Wrist-worn devices tend to be unobtrusive and acceptable to the public, but wrist motion is relatively limited for eating detection and we expect it to be difficult to achieve high accuracy, especially in free-living scenarios.

We chose to place the sensor behind the ear, right on the tip of mastoid bone (Figure 2.1);

Figure 2.1: Tip of mastoid bone

this location has been shown to give a stronger chewing signal to a contact microphone than other locations on the jaw or neck [55]. In addition, a device placed behind the ear does not impede hearing, unlike earbuds or ear-canal sensors. Moreover, this location, once the device is miniaturized, may allow a user to wear the device privately, i.e., other people would not see it and would not know that it is there (as in modern hearing aids).

2.2 Approach

To evaluate our method in Section 2.3, we developed two apparatus: a bench-top apparatus and a wearable apparatus.

2.2.1 Bench-top apparatus

We evaluated two off-the-shelf sensors for a behind-the-ear device: a contact microphone (CM-01B, Measurement Specialties) and an EMG sensor (AT-04-001, MyoWare Muscle Sensor). The microphone uses a PVDF piezo film combined with a low-noise electronic preamplifier to pick up sound applied to the central rubber pad; a metal shell minimizes external acoustic environmental noise. The 3 dB bandwidth of the microphone ranges from 8 Hz to 2200 Hz, and covers our frequency range of interest. Both sensors are connected to a data acquisition device (DAQ) (USB-1608G, Measurement Computing) with a 20 kHz sampling rate and a 24-bit resolution, while the data collected is processed and analyzed on

Figure 2.2: Contact microphone

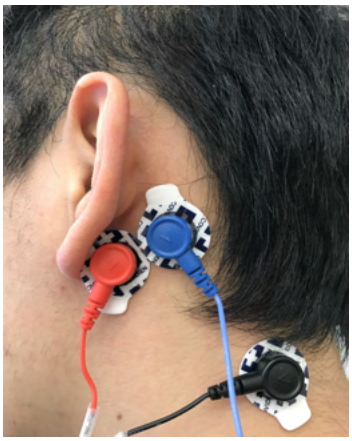

Figure 2.3: EMG electrodes

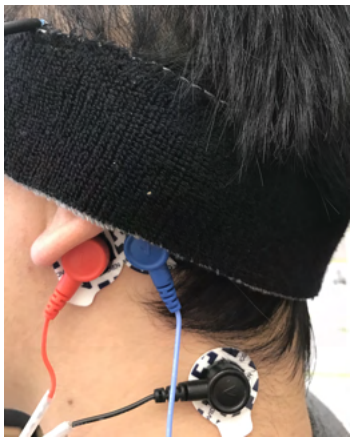

Figure 2.4: Experiment setup

a laptop.

As shown in Figure 2.2, the location we used for acoustic sensing is the tip of mastoid bone, a relatively hard surface behind the ear. We fixed the contact microphone under a headband during data collection to maintain stable contact with the body. For the EMG sensor, we used three Ag/AgCl electrodes with gel (24mm in diameter), placed as shown in Figure 2.3. The ground electrode can be placed anywhere on the body as long as it is relatively far away from the other two electrodes. For convenience, we placed the ground electrode on the back of participants' necks. Figure 2.4 shows an experiment setup where both sensors are attached to a participant.

2.2.2 Wearable apparatus

In addition, we developed a wearable device (Figure 2.5), which is also a preliminary prototype of *Auracle* (about which there are more details in Chapter 3). Based on the results in Section 2.4.2, we chose to incorporate only a microphone in our wearable device. We fused the contact microphone, MCU (ATSAMD21, SparkFun), SD card and 400 mAh battery into a headband. For this preliminary prototype, we expect the battery life to be at least 8 hours.

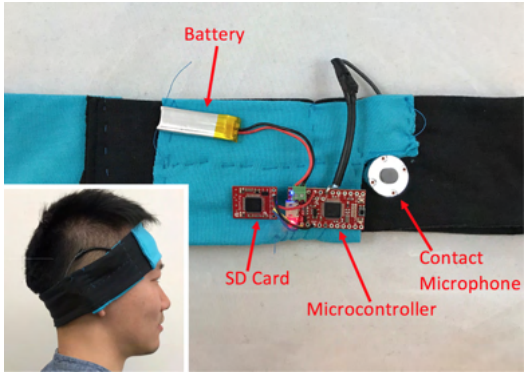

Figure 2.5: Wearable apparatus

2.3 Method

Our experiments involved multiple stages, including data collection on the bench-top apparatus, feature extraction, feature selection, and classification on a laptop.

2.3.1 Data Collection

With the approval of our Institutional Review Board (IRB), we collected data from 20 participants (8 females, 12 males; aged 21-30). For the first 10 participants, we collected data using both contact microphone and EMG sensors. Based on the experiments with the first 10 participants (Section 2.4.2), we concluded that the EMG sensor was infeasible for free-living scenarios and provided only limited improvement to the accuracy of eating detection. We thus collected data from the second 10 participants using only the contact microphone. All the activities listed in Table 2.1 were performed, in sequence, by each participant. The total duration of both positive cases (*Eating*) and negative cases (*Non-eating*) are each 12 minutes. All participants ate the same six types of food, shown in Figure 2.6, among which three (protein bars, baby carrots, crackers) are crunchy while the other three (canned fruits, instant foods, yogurts) are soft. While recording each activity, participants were asked to refrain from performing any other activity and to minimize the gaps between each mouthful. All data recorded during each activity was labeled as the activity.

Figure 2.6: Six types of food used for data collection

Activity	Description	Duration
Eating	Eat a protein bar	2 minutes
Eating	Eat several baby carrots	2 minutes
Eating	Eat several crackers	2 minutes
Eating	Eat canned fruit	2 minutes
Eating	Eat instant food	2 minutes
Eating	Eat yogurt	2 minutes
Talking	Read an article aloud	5 minutes
Silence	Relax and avoid chewing	5 minutes
Coughing	Cough	24 seconds
Laughing	Laugh	24 seconds
Sniffling	Sniffle	24 seconds
Deep Breathing	Deep breath	24 seconds
Drinking	Drink water	24 seconds

Table 2.1: The list of activities performed by each participant for data collection

2.3.2 Feature Extraction and Selection

As sampling rate is one of the most important factors driving power consumption for wearable sensors, we hoped to use a relatively low sampling rate. After testing a range of sampling rates from 250 Hz to 4000 Hz, we chose 500 Hz for eating detection in our system. As a result, all raw data was first downsampled from 20 kHz to 500 Hz before feature extraction. Since the frequency of non-speech body sounds is generally higher than 20 Hz [55], we used a high-pass filter to minimize the frequency components lower than 20 Hz. The filtered signals were segmented into time windows with uniform length and 50% overlap. In this work, we experimented with window sizes ranging from 1 second to 5 seconds and the results are shown in Figure 2.8. For each time window, we used the open-source Python package *tsfresh* to extract a common set of 206 features per sensor from both time and frequency domains.

The two sensors provide a total of 412 features for evaluation. To improve computational efficiency, we selected relevant features based on feature significance scores and the Benjamini-Yekutieli procedure [10]. Each feature is individually and independently evaluated with respect to its significance for predicting the target under investigation and a p-value is generated to quantify its significance. Then, the Benjamini-Yekutieli procedure evaluates the p-value of all features to determine which ones to keep.

2.3.3 Classification

We designed a two-stage classification model. In the first stage, to filter out most of the time windows labelled as *silence* using simple thresholding, we calculate the average variance of all time windows labelled as *silence* by ground truth, and find all time windows with lower variance in the entire data set and mark them as "evident silence periods". After separating training and testing data, we train our classifier on the training set excluding the "evident silence periods". Similarly, during testing, we arbitrarily mark the time windows in the testing set that are "evident silence periods" as *Non-eating*. To reduce energy consumption,

when we implement a compact, low-energy device, the first stage classification can be done in hardware so that the device does not need to process data during the "evident silence periods". In the second stage, we choose a Logistic Regression classifier to perform a 2-class classification to classify *Eating* and *Non-eating*. We chose Logistic Regression as it is lightweight enough to be implemented in a resource-limited wearable. In both the training and testing data sets, *Eating* is one class and all other seven activities are treated as another class, *Non-eating*.

2.4 Evaluation

We evaluated our methods with varying sensors, window sizes, bit resolutions, and number of features. We also conducted an uncontrolled-food experiment.

2.4.1 Evaluation metrics

To evaluate the accuracy of our classifier, we compared its output for each time window against the ground-truth label for that time window. In other words, each time window is an independent test case that results in one of four outcomes:

True positive (TP): Both the classifier and ground truth indicate *Eating*.

False positive (FP): The classifier indicates *Eating* and ground truth indicates *Non-eating*.

True negative (TN): Both the classifier and ground truth indicate *Non-eating*.

False negative (FN): The classifier indicates *Non-eating* and ground truth indicates *Eating*.

We then evaluate our method with three metrics:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

The accuracy score is balanced as we configured our data to be 50% positive cases (*Eating*) and 50% negative cases (*Non-eating*).

We used Leave-One-Person-Out (LOPO) cross-validation to evaluate our classifier's performance. A LOPO model is relatively unbiased because the classifier is asked to detect eating for a new person whom it has not seen before. The model iterates over all possible combinations of the training and testing data set. For each iteration, the data set is divided into two subsets: the testing set (data from one participant) and the training sets (data from all other participants). The classifier is trained on the training sets and outputs three metrics (accuracy, precision, and recall) on the testing set for each iteration. As summary metrics, we calculated the mean and standard deviation of these three scores across all iterations.

2.4.2 Sensor Comparison

Figure 2.7 shows the results of eating detection with contact microphone and EMG, independently and combined, for the first 10 participants. During our experiments, we found that repeatable and effective placement of the electrodes used for collecting EMG signals was a challenging task and participants found this task to be unpleasant. Moreover, Figure 2.7 shows that EMG and contact microphone improve accuracy by 3.2% (with a p-value of 0.005) relative to use of the contact microphone alone. Although statistically significant, this difference is not great enough to be worthwhile given the extra cost, effort, and size that would be incurred. EMG also appears to yield the worst performance on its own. We thus decided that it is infeasible to integrate EMG sensors into a wearable suitable for free-living scenarios. In the final 10 participants, we collected data using only the contact microphone and used data from the contact microphone alone for evaluation in Sections 2.4.3 and Section 2.4.4.

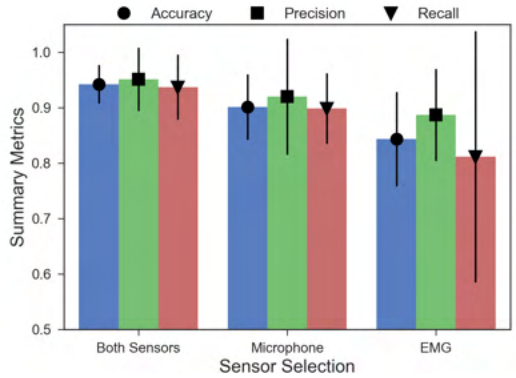

Figure 2.7: Summary metrics when using contact microphone and EMG, independently and combined (error bars represent standard deviation).

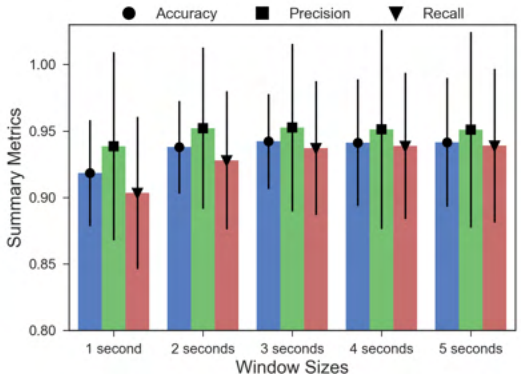

Figure 2.8: Summary metrics when window size ranges from 1 second to 5 seconds (error bars represent standard deviation)

2.4.3 Parameter evaluation

We explored the effect of different window sizes on accuracy of eating detection in our system by testing a range of window sizes from 1 second to 5 seconds. Based on the accuracy results shown in Figure 2.8, we chose a 3-second window size for all later evaluations as it yielded the best accuracy.

Moreover, we evaluated whether the bit resolution of analog-to-digital converters (ADC) affects the classification performance. We rounded our raw data (in decimal form) to the third decimal place before feature extraction to simulate 10-bit resolution ADC in a 1V voltage range. As shown in Table 2.2, lowering the bit resolution did not have a substantial effect on the accuracy of eating detection, so we used a 10-bit resolution for later evaluation.

Resolution	Accuracy	Precision	Recall
24-bit	**0.942**	**0.953**	**0.937**
	± 0.036	**± 0.063**	**±0.050**
10-bit	0.935	0.943	0.934
	± 0.043	± 0.075	±0.052

Table 2.2: Results when bit resolution was 24-bit and 10-bit (mean value ± standard deviation)

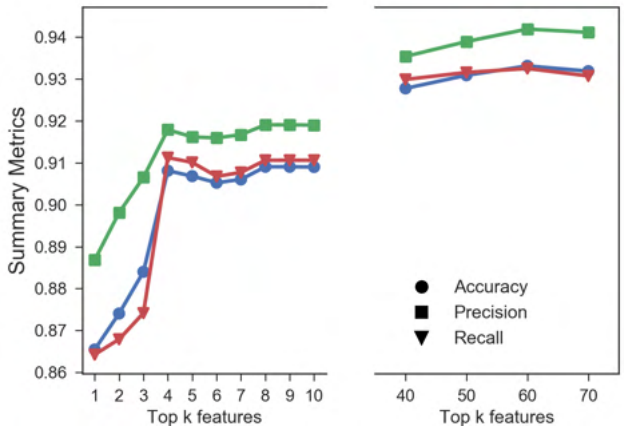

Figure 2.9: Results when number of features ranged from 1 to 70

Finally, considering the limited computational resources of wearable platforms, we further selected a smaller number of features using the Recursive Feature Elimination (RFE) algorithm with a Lasso kernel. Figure 2.9 shows the results when the number of top features ranged from 1 to 70.

In general, an increased number of features can benefit accuracy but the improvement is limited (the largest difference of accuracy was less than 8%). To achieve a relatively high accuracy and avoid overfitting due to insufficient features, we chose the top 8 features for later evaluation (Table 2.3). When we only used the top 8 features for classification, the accuracy, precision, and recall metrics were 90.9%, 91.9%, and 91.1% respectively.

Feature type	Description	Number
Kurtosis	Kurtosis	1
Mean	Number of values higher than mean	1
Sum	Sum over the absolute values of changes	1
Peak	Number of peaks at different width scales	4
Friedrich coefficients	Coefficients of polynomial h(x) fitted to the deterministic dynamic of Langevin model [27]	1

Table 2.3: Top 8 features

Figure 2.10: Food brought in by the participant

2.4.4 Uncontrolled-food evaluation

To further evaluate our system on food that was not used for training and under a more realistic condition, we designed an uncontrolled-food experiment. First, using the acoustic data collected from all 20 participants, we trained a classifier with the top 8 features (Table 2.3) extracted using the same methods as described in Section 2.3.2. Then, using the bench-top apparatus described in Section 2.2.1, we asked one participant to conduct a sequence of activities and used the pre-trained classifier to classify these activities in real time with the same classification methods as described in Section 2.3.3. The food was brought in by the participant (Figure 2.10) and not like the food used in the training data. To conveniently annotate activities for the ground truth, we asked the participant to perform a series of activities lasting 30 or 15 seconds each following an arbitrarily predetermined routine. The total time length of each type of activity performed in the routine is shown in Table 2.4.

Activity	Number of periods	Total time length
Eating	10	300 seconds
Talking	4	120 seconds
Silence	6	105 seconds
Coughing	1	15 seconds
Laughing	1	15 seconds
Sniffling	1	15 seconds
Deep Breathing	1	15 seconds
Drinking	1	15 seconds

Table 2.4: Activities performed in uncontrolled-food evaluation

The accuracy, precision, and recall metrics for this experiment were 91.5%, 95.1%, and 87.4%. These results show that our system can work properly when participants eat food that was not used for training the classifier.

2.5 Related work

Below is a brief overview of existing methods evaluated in laboratory conditions, which we categorize into two types: *acoustic* and *other*. Acoustic approaches can be further classified as *air* (using microphones designed for recording sound from the air) and *contact* (using microphones designed for recording sounds conducted through the body). For the second type, these microphones typically require direct contact with the skin.

Air-conducted sound: Amft et al. evaluated the air-conducted sound intensity of chewing and speech when a microphone is placed at different locations on the body [1]. They identified the optimal location to be the inner ear, directed towards the eardrum, rather than 2 cm in front of mouth, at the cheek, collar bone, behind the outer ear or 5 cm in front of the ear canal opening. Since then, much effort has been put in developing ADM systems using air microphones positioned in the ear [39, 49, 61]. Sazonov et al. explored the option of using the neck as the sensing locations and achieved 84.7% average weighted accuracy in detection of swallowing events [61, 62].

Body-conducted sound: To capture and recognize a diverse range of body-conducted sounds, including eating sounds, Rahman et al. designed a mobile sensing system consisting of a customized contact microphone placed on the neck, an ARM MCU and Android smartphone [55]. They achieved an average recall of 71.2% for a nine-class classification of different body sounds (eating, drinking, deep breathing, clearing throat, coughing, sniffling, laugh, silence, speech) in laboratory conditions. Several other acoustic-based ADM systems also used body-conducted sound recorded from the neck [46, 61, 85] or in the ear canal [67] to detect swallowing or chewing events.

Compared with normal air microphones, contact microphones capture internal vibrations directly from the body surface and are naturally immune to ambient noise, making these sensors promising for eating detection in out-of-lab, free-living scenarios, where ambient noise is variable and can be large in magnitude. Because we are most interested in detecting eating in free-living scenarios, Auracle was designed with a contact microphone as the eating-detection sensor.

Other eating-detection approaches evaluated in laboratory conditions include physiological, piezoelectric, proximity and fusion approaches. The two primary physiological signals explored for eating detection include electroglottography (EGG) and EMG. EGG sensors capture the motion-induced variations of electrical impedance recorded between two electrodes positioned on the larynx [36]. Farooq et al. placed an EGG setup around participants' necks to capture swallowing events and achieved an average per-epoch classification accuracy of 90.1% [23]. Zhang et al. fused three EMG electrodes into an eyeglasses frame to capture muscle signals during eating [87, 90]. Using dry fabric electrodes, they could detect chewing with a precision and recall of 80%. Piezoelectric sensors can produce a voltage at their terminals in response to mechanical stress [30]. To automatically monitor eating behavior, piezoelectric film sensors were placed on the jaws [24] or throat [30] for motion capture. Kalantarian et al. developed a necklace to capture swallowing events [30] and were able to detect more than 81.4% of swallows. Finally, many systems fused two or more of

these approaches with the aim of improving automatic intake monitoring systems [19,41,48]. Merck et al. presented a multi-sensor study of eating recognition, which combines head motion, wrist motion and audio [41]. In their study, using audio sensing alone achieved 92% precision and 89% recall in finding meals, while motion sensing was needed to find individual intakes.

2.6 Summary

In this chapter, we built a data-acquisition system using two off-the-shelf sensors and collected data of 8 activities from 20 participants. In LOPO cross-validation experiments, we achieved accuracy over 90.9% with 500 Hz sampling rate, 10-bit resolution, 3-second window size and 8 features for eating detection of 6 types of food with different crunchiness level (3 crunchy and 3 soft). Additionally, we experimented with a wearable apparatus and showed promising preliminary results.

3

Eating detection in free-living conditions

Despite substantial research on technology for automatic eating recognition [1,5,7,11,30,43,48,88,90,91], the most common method is still manual record-keeping. It has not yet been possible to accurately and automatically detect eating outside the lab; thus our interest is to make a wearable system robust enough for free-living scenarios. To move from laboratory environments to "the wild," there are two major challenges. First, in out-of-lab settings, a variety of environmental noise and subject activities could be misclassified as eating, including but not limited to the seven non-eating activities discussed in Section 2.3.1. Second, it is challenging to build a system that is energy-efficient, unobtrusive and comfortable for different individuals to wear for an entire day.

To automatically recognize eating in free-living conditions, we designed and built the *Auracle* wearable eating-recognition system. As in Chapter 2, we assume that chewing is a first-level indicator of eating activity, so Auracle uses a contact microphone mounted behind the ear to detect chewing sounds.

Regarding the work described in this chapter, I made the following contributions:

- I coordinated the development of hardware, software and mechanical design of Auracle, the first ear-mounted system using a contact microphone for eating detection in free-living conditions.
- I developed a privacy-preserving approach for obtaining ground-truth information about eating behavior in free-living conditions.
- I collected field data with 14 participants for 32 hours in free-living conditions and additional eating data with 10 participants for 2 hours in a laboratory setting.
- I implemented a data-analysis approach and a classifier that achieved accuracy exceeding 92.8% and F1 score exceeding 77.5% for eating detection. Moreover, I developed a two-stage aggregation approach and detected 20-24 eating episodes (depending on the metrics) out of 26 in free-living conditions.
- I estimated the power consumption of Auracle when running in three different modes.

Regarding the work described in this chapter, I acknowledge the contributions of others:

- Tao Wang implemented feature extraction and classification algorithms on the MCU and assisted with the implementation of the data-analysis approach.
- Nicole Tobias designed the Auracle printed circuit board (PCB).
- Josephine Nordrum implemented data-logging functions on the MCU.
- Shang Wang assisted in the data collection and implementation of the data-analysis approach.
- George Halvorsen developed and 3D-printed the Auracle mechanical housing.
- Ronald Peterson implemented Bluetooth Low Energy (BLE) functions on the MCU.

For these above-mentioned contributions, I provided collaborative input.

3.1 Background

As in the previous chapter, we define *eating* as "an activity involving the chewing of food that is eventually swallowed." For our work in this and following chapters, we define an *eating episode* as: "a period of time beginning and ending with eating activity, with no internal long gaps, but separated from each adjacent eating episode by a long gap, where a *gap* is a period in which no eating activity occurs, and where *long* means a duration greater than a parameter δ." We chose $\delta = 15$ minutes in our studies as suggested by Leech et al. [37]. We used this definition for the episode-based evaluations in Section 3.5.2.

Although several researchers have designed systems that use various cues to determine eating (e.g., audio information from the ear canal [1, 39, 49, 61, 67] or throat [46, 55, 61, 62, 85], first-person or third-person images [56, 69, 73], wrist-based gesture recognition [22, 64]), these systems have practical limitations. They are either obtrusive (microphone on throat), uncomfortable (bulky), privacy invasive (images capturing other people) or unnatural (wearing a watch on the dominant hand).

Unlike these prior approaches, our head-mounted design is similar to a behind-the-head earphone and is comfortable to wear in everyday settings. Our device can infer eating episodes, in real-time, on the wearable device, and log these events as they occur, or opportunistically alert a smartphone or smartwatch about detected eating behaviors.

We have seen only a few wearable eating-recognition systems that be used in free-living settings, and can run on-board, real-time feature-extraction and classification algorithms; such on-board processing can decrease the recognition latency and better protect user privacy. Auracle can locally capture, process, and classify sensor data collected in out-of-lab, day-long, free-living scenarios.

Similarly, prior eating-detection research has traditionally worked to detect eating without considering energy efficiency and battery life – both critical factors for improving the size, weight, comfort, cost, and usability of any wearable device. To develop an eating detection system that works well beyond carefully-controlled laboratory settings, we devel-

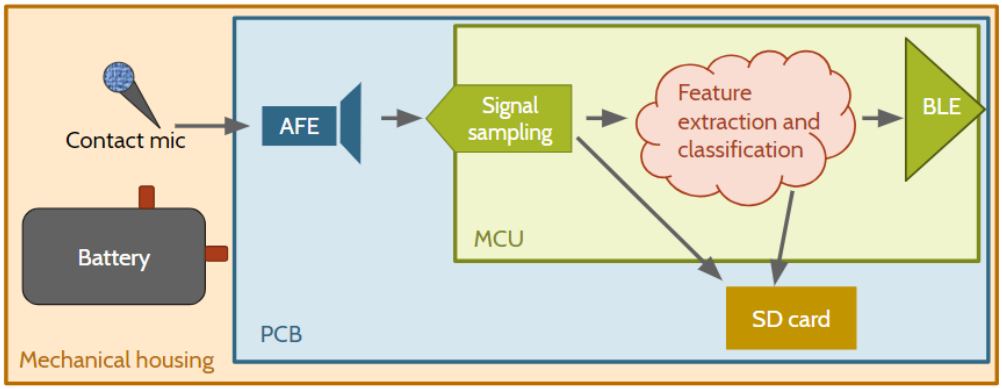

Figure 3.1: Auracle prototype

oped Auracle using an ultra-low-power MCU (Section 3.2.3), evaluated Auracle's energy efficiency (Section 3.6) and estimated that our prototype can monitor eating continuously while lasting 28.1 hours when paired with a 110 mAh rechargeable battery.

3.2 System design

The Auracle system (shown in Figure 3.1) includes a contact microphone (Figure 3.2), a battery, a custom-designed PCB for data acquisition, and a wearable mechanical housing. The PCB (Figure 3.3) incorporates an Analog Front End (AFE) for signal amplification, filtering, and buffering, an MCU for signal sampling and processing, feature extraction, eating activity classification, and system control, a Bluetooth radio for data transmission, and a micro-SD card socket for long-term data storage. The signal and data pipeline from the contact microphone includes AFE-based signal shaping, MCU-based analog-to-digital conversion, on-board feature extraction and classification, and data transmission and storage. We implemented data-logging functions to write raw data, feature values or prediction results to the SD card for our research. We also implemented BLE functionality in the MCU so the Auracle prototype can also transmit these data through BLE, if needed. The total cost of the current prototype, including PCB fabrication and component costs, is $80 per unit, and would drop to $66 if ordered in quantities of 1,000 or more.

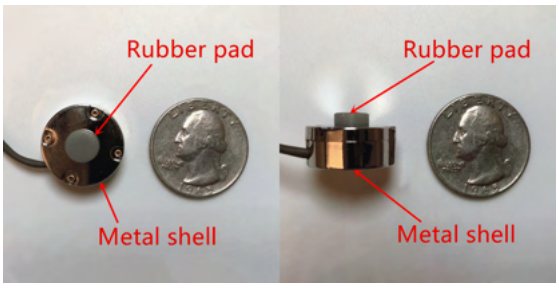

Figure 3.2: Contact microphone

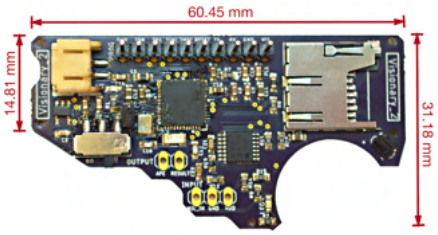

Figure 3.3: Auracle's PCB Design

We developed Auracle in three stages. In Stage I, we built three prototypes and used them for acquiring field data (Section 3.3.1) and additional eating data (Section 3.3.2). We implemented only the functions required for data acquisition on the MCU. We analyzed the data (Section 3.4) and evaluated eating-detection performance (Section 3.5) offline on a laptop. In Stage II, we implemented on-board feature extraction and classification based on the most promising features (Table 3.1) and classification models determined in Stage I. We trained the classification model (Section 3.4.3) offline on a laptop using the in-lab and field data recorded (Section 3.3.1 and 3.3.2); the classification model was then implemented in embedded-C and ported to the MCU. The on-board classification model uses the feature values extracted from windows of audio samples as inputs to classify windows as periods of *eating* or *non-eating*. In Stage III, we added a Bluetooth radio on our PCB and implemented the BLE functionality in the MCU, which could be used to provide users with real-time interventions.

3.2.1 Contact Microphone

We used the same contact microphone introduced in Section 2.2 to capture chewing sound (shown in Figure 3.2). According to the data sheet, when powered by 3.3V the power consumption of the microphone is 0.33 mW. This microphone has been used in electronic stethoscopes and, based on preliminary studies in Chapter 2, we found it to be sufficiently

sensitive to detect chewing sounds.

3.2.2 Analog Front End

To make the most of the MCU's analog-to-digital converter's (ADC) input dynamic range, the contact microphone signal is conditioned by an AFE. The AFE level-shifts the contact microphone signal, amplifies it by 15 dB, and bandlimits it to the 20–250 Hz frequency range. We chose the frequency range and amplification gain based on the experiment results from Chapter 2.

3.2.3 Microcontroller Unit

An embedded MCU samples the output signal from the AFE, processes data, and communicates results. A 500 Hz sampling rate with 10 bits of resolution is required to sample typical eating signals from a contact microphone [11]. To meet these requirements, the Auracle prototype employs a Texas Instruments (TI) CC2640R2F Simplelink Wireless MCU (ARM Cortex M3) with an integrated sensor controller and BLE module. The MCU samples and stores data over a 3-second window to construct a 1500-value array from which features are extracted and classified as *eating* or *non-eating* events. The MCU can record to the SD card raw data, summary data (i.e., feature values or prediction results), or both, depending on operating mode. The MCU can also transmit these data through BLE, if needed. The Auracle application leverages TI's operating system (TI-RTOS) for simplified task threading and automatic low-power optimization. We developed programs for the main CPU in TI's Code Composer Studio and designed and generated the firmware image for the Sensor Controller using TI's Sensor Controller Studio.

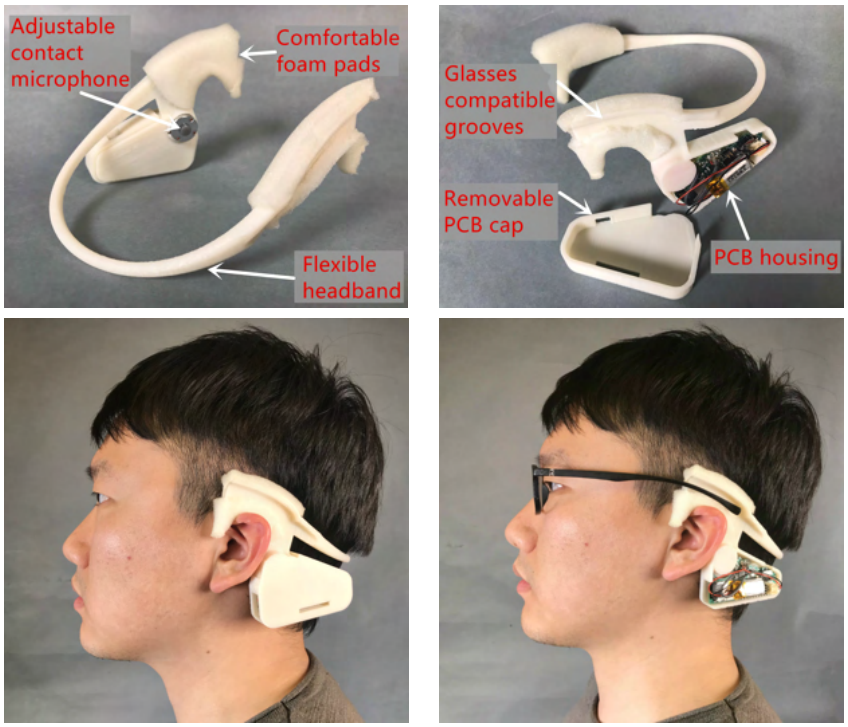

Figure 3.4: Mechanical housing of Auracle

3.2.4 Printed Circuit Board

The Auracle prototype hardware integrates a custom PCB housed in a 3D-printed head-mounted plastic enclosure, detailed below. Figure 3.3 shows the PCB, which comprises the CC2640R2F MCU, a 110 mAh battery, the contact microphone (Section 3.2.1), a Bluetooth radio, a micro-SD card socket, and the custom AFE (Section 3.2.2). Our PCB implementation is small enough to be deployed in free-living conditions and its unique shape was designed to fit within the wearable form-factor of the head-mounted housing. The semicircular arc was added to the PCB design to provide a structured fit for the contact microphone.

3.2.5 Mechanical Housing

The Auracle enclosure consists of a 3D-printed ABS plastic frame that wraps around the back of a wearer's head and houses the PCB, battery, and contact microphone (Figure 3.4).

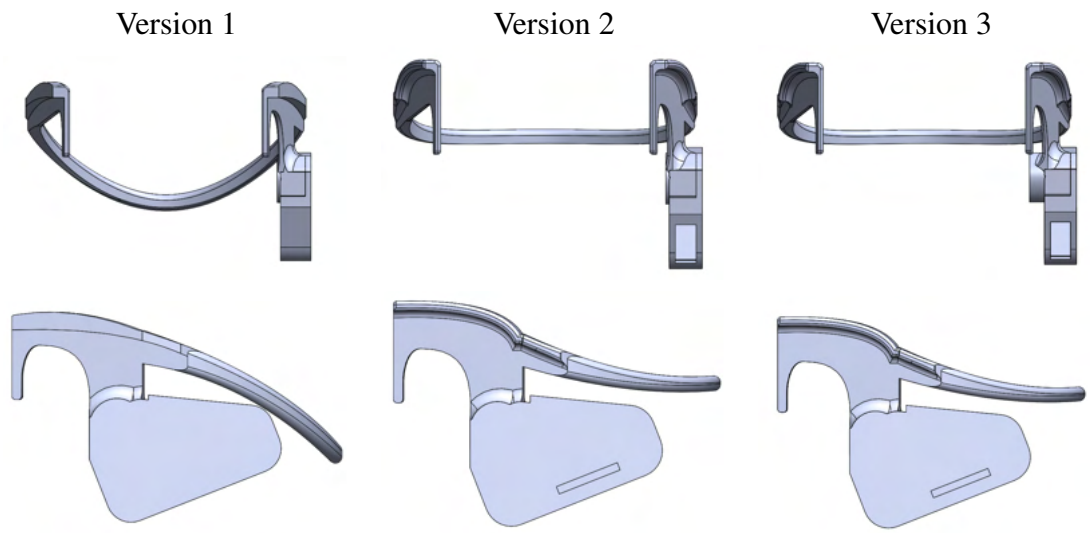

Figure 3.5: Three versions of mechanical housing design

Soft foam supports the enclosure as it sits above a wearer's ears. There are grooves in the enclosure making Auracle compatible with most types of eyeglasses. The contact microphone is adjustable, backed with foam that can be custom fit to provide adequate contact on different head shapes. This adjustment is necessary because Auracle is built on the premise that the contact microphone has proper contact with the mastoid bone. An adjustable microphone mount ensures that Auracle can cater to several head shapes and bone positions.

There are three versions of the enclosure to fit various head shapes (Figure 3.5). Version 1 wraps lower around the head than Versions 2 and 3. Version 3 has an extra extrusion to hold the contact microphone closer to the wearer if their mastoid bones are more recessed relative to their ears. All versions are 12.7cm $\times$ 12.7cm $\times$ 8.6cm.

3.3 Data collection

Using sensor data recorded with Auracle in both field and laboratory settings, we determined an optimal set of features and an appropriate classification algorithm to implement in the digital back-end running on our PCB. We also used these data as training data for our

classification model, and to derive the performance (Section 3.5) in terms of accuracy and power-consumption evaluation (Section 3.6). Under a protocol approved by our Institutional Review Board (IRB), we collected data in both free-living scenarios and a laboratory setting.

3.3.1 Field Data Collection

Auracle is aimed at use in free-living conditions, so we conducted a field study with 14 participants. The goal of this study was to collect raw audio data for the purpose of *developing* and *evaluating* the Auracle itself, as noted above. To do so, we had to address a critical challenge – we need a reliable way to obtain "ground truth" in free-living conditions. In short: *when* did the participants actually eat?

We thus developed an approach for ground-truth measurement. It is important to note that this mechanism is not part of the envisioned use of Auracle – just part of its development. We fused an off-the-shelf wearable miniature camera into a baseball cap and used the camera to record video during the field studies (Figure 3.6). The camera was fixed under the brim of the cap and directed at the mouth of the participants only; this orientation made it difficult to identify the participant by watching videos and also avoided recording anyone else, other than the study participant. The ambient microphone built into the camera was physically removed before the study so no audio would be captured. All the videos recorded during the study were stored in an SD card for later annotation. Compared with other similar apparatus [5], our ground-truth collector is relatively unobtrusive. Figure 3.7 shows two screen shots of the video recorded by the camera during eating and non-eating periods, respectively. Again, the ground-truth collector is not part of the operational Auracle and was used just for development and evaluation.

Field Studies

We collected data from 14 participants (2 females, 12 males; aged 20-33; 10 wore glasses; 2 had long hair). These participants were mostly college students and staff. For each

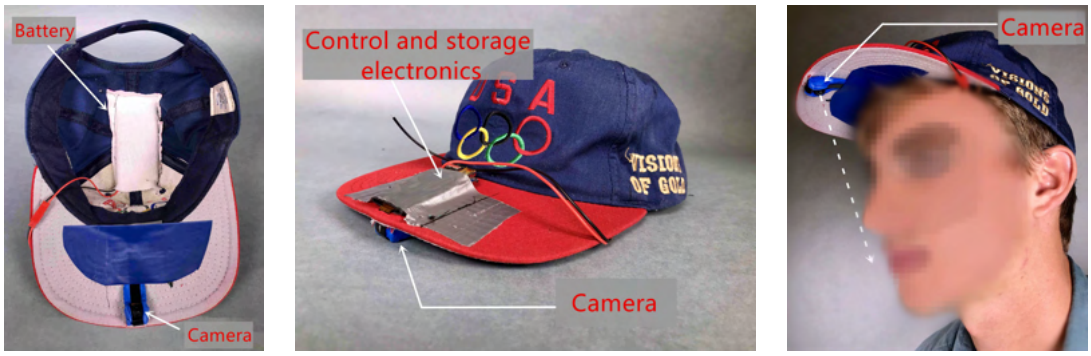

Figure 3.6: Ground-truth collector

Figure 3.7: Screen shots of the video recorded by ground-truth collector during eating and non-eating periods

session, the participant was compensated with a $20 gift card. Among the 14 participants, 12 participants chose to participate in 1 session of the study while 2 participants chose to participate in 2 sessions. Each session lasted 2 hours. Overall, we collected a total of 32 hours of field data. After preliminary review, we found 2 sessions (4 hours) of the field data, collected from 2 different participants, could not be used for further analysis. In one session, the video recorded by cap-mounted camera was totally blocked by the participant's nose, making it hard to determine whether the participant was eating. In another session, the contact microphone signal was too weak due to poor contact and barely changed during session. We excluded the data collected during these two sessions. We used the remaining 28 hours of data recorded from 12 participants for analysis (Section 3.4) and evaluation (Section 3.5). During these 28-hour periods of field data acquisition, participants ate various types of food including rice, bread, noodles, meat, vegetables, fruit, eggs, nuts, cookies, crackers, soup and yogurt. Participants recorded data in diverse environments including

houses, offices, cars, restaurants, dining halls, kitchens, and streets.

Before the start of each session, the participant was asked to wear the Auracle prototype in Stage I (Section 3.2) and the ground-truth collector (Figure 3.6). To ensure the contact microphone in our prototype had good contact with mastoid bone (Figure 2.1), we first visually inspected whether the central rubber pad of the contact microphone remained in contact with the skin when the participant turned her or his head back and forth. We then asked the participant to stay silent for 10 seconds, followed by chewing a baby carrot for another 30 seconds. If the amplitude of the data recorded during the chewing period was larger than that in silent period, we concluded there was good contact between the microphone and skin.

At the beginning of each session, we asked the participant to tap on their cheek and the mechanical housing of the prototype using their hands three times, which could be recorded by both head-mounted camera and Auracle. We then asked the participants to go about their normal daily activities outside the lab. Their behavior and location were uncontrolled, but the participants were asked to wear the Auracle and the cap continuously during their time in the field. Also, we requested that at least one eating episode take place at anytime during the session. At the end of the session, we asked the participant to perform the same three-tap event. We used these three-tap events at the beginning and end of the session to synchronize the video and audio data collected. A example of one session of field data collection is shown in Figure 3.8, where the parts in black boxes represent eating periods.

Video Annotation

To annotate the videos (i.e., labeling moments as *eating* or *not eating*), we used the video annotation service from Baidu.[1] We uploaded all field study videos to the Baidu Drive for review. Three Baidu annotators independently watched and annotated the periods of eating in each video, with 1-second resolution.

[1] http://zhongbao.baidu.com/

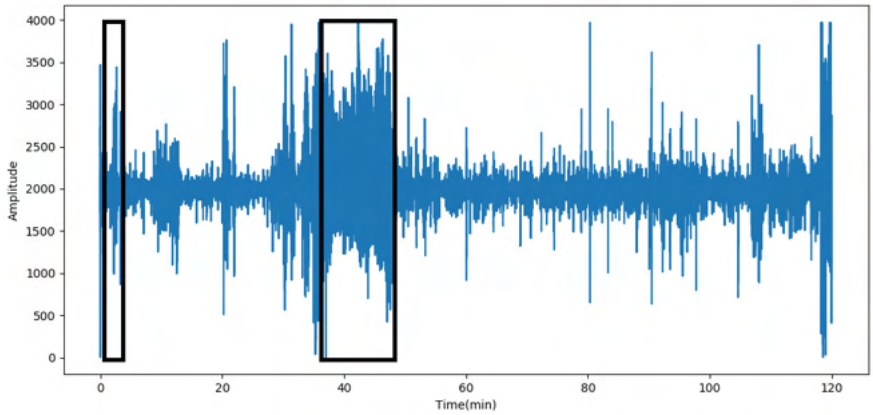

Figure 3.8: Temporal signature of one session of field-data collection (black boxes indicate periods of eating)

We calculated the proportion of the annotation-mismatch periods across each of the 3 reviews. Each 1-second window over which the three annotators disagreed were defined as annotation-mismatch periods. The proportion of the annotation-mismatch periods in 14 the field-study videos was small (mean: 2.79%; standard deviation: 1.85%). Thus we concluded all the videos were annotated carefully by three annotators.

We converted the three annotation results into a single label file used for experiments in Section 3.4 and 3.5. The label file was generated based on the majority annotation results from three annotators. For example, if two or more annotators annotated a 1-second period of video as *eating*, it was labeled *eating* in the final label file; otherwise it was labeled *non-eating*.

Finally, since our predictions were based on 3-second windows, we converted the resolution of the labeling result from 1 second to 3 seconds. We found that there were very few 3-second windows (less than 0.78%) that contained both *eating* and *non-eating* labels. We labeled a 3-second window *eating* if it contains any *eating* labels within the window; otherwise we labeled that window *non-eating*.

3.3.2 Additional Eating-data Collection

Since the data collected in free-living scenarios is unbalanced (i.e., much less time spent on *eating* than *non-eating*), we collected additional in-laboratory eating data to augment the training dataset. The additional data allowed us to explore whether the addition of in-laboratory eating data would improve the classification results (Section 3.5.2).

We collected data from 10 participants (2 females, 8 males; aged 21–33; 8 wore glasses; 2 had long hair) in the laboratory condition. At the start of each session, each participant was asked to wear the Auracle prototype described in Section 3.2. We used the same visual and data inspection methods used (Section 3.3.1) to verify Auracle placement in this cohort.

We asked the participants to eat six different types of food, one after the other. The food items (Figure 2.6) included three crunchy types (protein bars, baby carrots, crackers) and three soft types (canned fruits, instant foods, yogurts). We asked the participants to chew and swallow each type of food for two minutes. During this eating period, participants were asked to refrain from performing any other activity and to minimize the gaps between each mouthful. After every 2 minutes of eating an item, participants took a 1-minute break so that they could stop chewing gradually and prepare for eating another type of food. A signal plotting of one entire session of lab data collection is shown in Figure 3.9, where the parts in black boxes represent eating periods. We removed data collected during the 1-minute break periods and concatenated all 2-minute eating periods into the additional eating dataset we used in Section 3.5.1.

3.4 Data analysis

In this section, we describe our evaluation metrics and multiple stages of our data processing pipeline (Figure 3.10) including data preprocessing, feature extraction, feature selection, classification, classification aggregation, and ground-truth label aggregation.

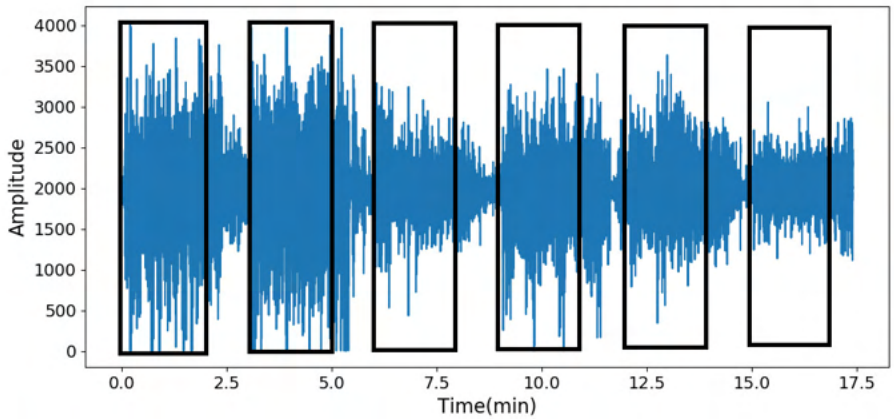

Figure 3.9: Temporal signature of one session of additional eating-data collection (black boxes indicate periods of eating)

3.4.1 Evaluation Metrics

We performed a Leave-One-Person-Out (LOPO) cross-validation to evaluate our classifier's performance in both *window-based evaluation* (described in Section 3.4.1) and *episode-based evaluation* (described in Section 3.4.1). A LOPO model is relatively unbiased because the classifier detects eating for a new person whose data it has not seen before. The model iterates over all possible combinations of the training and testing data set. For each iteration, the data set was divided into two subsets: the testing set (data from one participant) and the training sets (data from all other participants). The classifier is trained on the training sets and outputs metrics on the testing set for each iteration; we then compute average metrics across all iterations. For the LOPO experiments using additional eating data (Section 3.5.1), we added the additional eating dataset (Section 3.3.2) to the training sets in each iteration.

LOPO Window-based Evaluation

To evaluate the accuracy of our classifier, we compared its output for each 1-minute time window against the ground-truth label for that time window. In other words, each time window was an independent test case that resulted in one of four outcomes:

True positive: Both the classifier and ground truth indicated *Eating*.

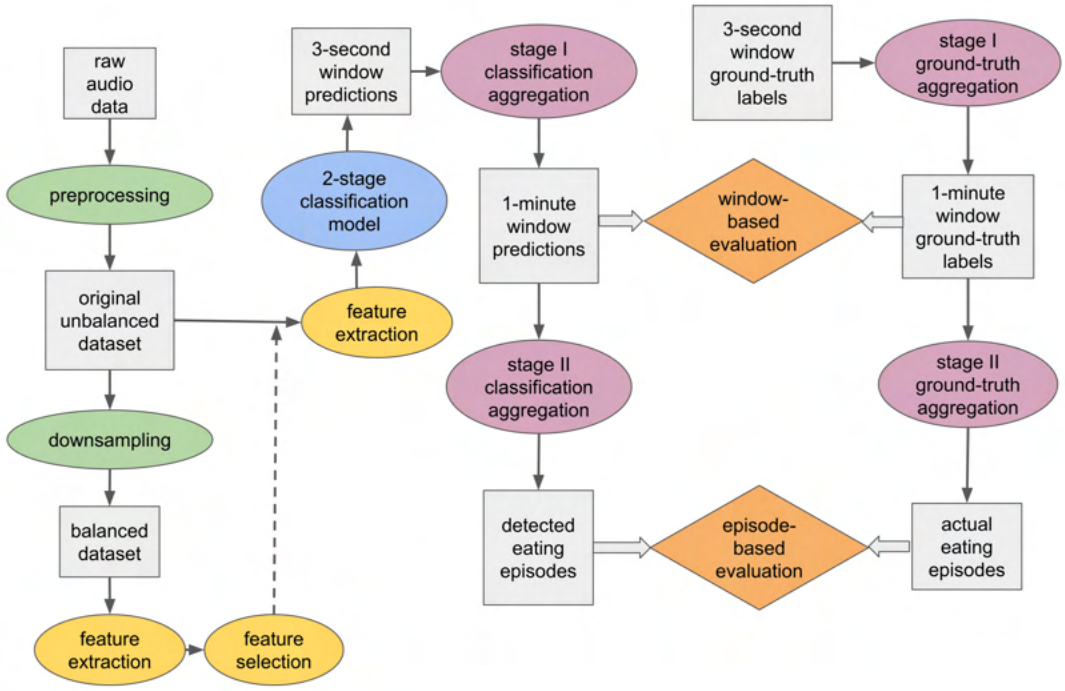

Figure 3.10: Data-processing pipeline

False positive: The classifier indicated *Eating* and ground truth indicated *Non-eating*.

True negative: Both the classifier and ground truth indicated *Non-eating*.

False negative: The classifier indicated *Non-eating* and ground truth indicated *Eating*.

We defined TP, FP, TN and FN as the number of true positive, false positive, true negative and false negative cases in the testing set, respectively. We then evaluated our method using five metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Weighted accuracy} = \frac{w \times TP + TN}{w \times (TP + FN) + FP + TN}$$

where w is the ratio of non-eating period vs. eating period; setting w = 1 yields Accuracy (non-weighted) [22, 48]. We set w in weighted-accuracy metrics based on the ratio of non-eating and eating period in the testing set for each LOPO iteration. As summary metrics, we calculated the mean and standard deviation of these five scores across all iterations. Using this evaluation method, each participant affected the summary metrics equally, regardless of whether they had 2-hour or 4-hour data recordings.

LOPO Episode-based Evaluation

We evaluated our method's ability to detect *eating episodes* using two metrics, the Jaccard similarity coefficient and the activity-recognition metrics proposed by Ward et al. [80].

Using an approach similar to previous work by Papapanagioto et al. [48], we matched each detected eating episode with either 0 or 1 ground-truth eating episode. We used the Jaccard similarity coefficient to determine whether this match led to a Correct Detection, False Detection, or Missed Detection.

Let the detected episode be represented as $E_d = [t_s, t_e]$, where t_s is the start of the detected eating episode and t_e is the end of the detected eating episode. Similarly, the actual eating episode (obtained from ground truth) is represented by $E_a = [t'_s, t'_e]$, where t'_s is the start of the actual eating episode and t'_e is the end of the actual eating episode. We can then define the Jaccard similarity coefficient as

$$J = \frac{E_a \ \cap \ E_d}{E_a \ \cup \ E_d}.$$

Each detected eating episode is an independent test case that results in one of three outcomes:

$$\text{Outcome} = \begin{cases} J \geq 0.55, & \text{Correct Detection} \\ 0 < J < 0.55, & \text{False Detection} \\ J = 0, & \text{Missed Detection} \end{cases}$$

For each Correct Detection, we also calculated the mean and standard deviation of the *delay* and *duration difference*. The *delay* is defined as the absolute value of the difference between the starting time of a detected and corresponding actual eating episodes. The *duration difference* is defined as the sum of the absolute value of the difference between the starting time and ending time of a detected and corresponding actual eating episodes.

Additionally, we evaluated our method using Ward's metrics. Ward et al. define an *event* as a variable-duration sequence of positive frames within a continuous time-series [80]. In our case, an eating episode represents an event and a 1-minute time window within the event represents a frame. An *event* can then be scored as either correctly detected (C); falsely inserted (I'), where there is no corresponding event in the ground truth; or deleted (D), where there is a failure to detect an event [80].

3.4.2 Data Preprocessing

As mentioned in Section 3.2.2, we first bandlimited signals to the 20–250 Hz frequency range using our AFE. The filtered signals were then segmented into non-overlapping windows of uniform duration. Based on some preliminary experiments testing a range of window sizes from 1 second to 5 seconds, we found that the 3-second window size gave us the best results so we chose 3 seconds as our default window size. Furthermore, because the signal amplitude was affected by the pressure applied to the contact microphone, which varied in each session due to different head shapes and microphone positioning, we used the RobustScaler function in Python's scikit learn package to normalize the data of each participant.

3.4.3 Feature Extraction and Selection

In our original field data set, the number of windows labeled as *non-eating* was significantly larger than the number labeled as *eating* (the time-duration ratio of data labeled as *non-eating* and *eating* is 6.92:1). When we selected features on this dataset, the top features returned provide us relatively good accuracy, but not always good recall and precision. However, recall and precision may be important metrics for some eating-behavior studies, so we first converted the original unbalanced dataset to a balanced dataset by randomly downsampling the number of *non-eating* windows so that we had equal number of *non-eating* windows and *eating* windows. We then performed feature extraction and selection on the balanced dataset (see Figure 3.10).

For each time window, we used the open-source Python package tsfresh[2] to extract a common set of 62 categories of feature from both time and frequency domains. Each feature category in this set can consist of up to hundreds of features when the parameters of the feature category vary. In our case, we extracted more than 700 features in total.

We then selected relevant features based on feature significance scores and the Benjamini-Yekutieli procedure [10]. We evaluated each feature individually and independently with respect to its significance in detecting eating, and generated a p-value to quantify its significance. Then, the Benjamini-Yekutieli procedure evaluated the p-value of all features to determine which ones to keep. After removing irrelevant features, considering the limited computational resources of wearable platforms, we further selected a smaller number of k features using the Recursive Feature Elimination (RFE) algorithm with a Lasso kernel ($5 \leq k \leq 60$). Table 3.1 summarizes the top 40 features.

Finally, we then extracted the same k features from the original unbalanced dataset to run the classification experiments ($5 \leq k \leq 60$).

[2]http://tsfresh.readthedocs.io/en/latest/

Table 3.1: Top 40 features selected by RFE algorithm

Feature category	Description	#features
FFT coefficients	Fourier coefficients of one-dimensional discrete Fourier Transform	29
Range count	Count of values within a specific range	1
Value count	Count of occurrences of a specific value	1
Number of crossings	Count of crossings of a specific value	3
Sum of reoccuring values	Sum of all values that present more than once	1
Sum of reoccuring data points	Sum of all data points that present more than once	1
Count above mean	Number of values that are higher than mean	1
Longest strike above mean	Length of the longest consecutive subsequence that is bigger than mean	1
Number of peaks	Number of peaks at different width scales	2

3.4.4 Classification

We designed a two-stage classification model to perform a binary classification on the original unbalanced dataset, using the set of features selected above. In Stage I, we used simple thresholding to filter out the time windows that seemed to include silence. We calculated the threshold by averaging the variance of audio data across multiple silent time windows. We collected this silent data during a preliminary controlled data-collection session. We identified time windows in the field data that had lower variance than the pre-calculated threshold and marked them as *evident silence periods*. After separating training and testing data, we trained our classifier on the training set excluding the *evident silence periods*. During testing, we labeled the time windows in the testing set that were *evident silence periods* as *non-eating*.

In Stage II, after experimenting with different commonly used classifiers without hyper-parameter optimizations (shown in Table 3.2), we chose a Logistic Regression (LR) classifier to perform a 2-class classification to classify *eating* and *non-eating* using the features we

Table 3.2: Results when using different classifiers with 40 features

Classifier	Accuracy	Precision	Recall	Weighted accuracy	F1 score
Logistics regression (LR)	0.928	0.757	0.808	0.879	0.775
K-nearest neighbors (K = 5)	0.888	0.621	0.810	0.858	0.689
Random forest	0.891	0.629	0.866	0.881	0.718
Decision tree	0.753	0.394	0.914	0.819	0.539
Gradient boosting	0.924	0.769	0.757	0.856	0.751

described in Section 3.4.2. We chose the LR classifier because it yielded the best F1 score in our experiment (shown in Table 3.2) and it is lightweight enough to be implemented in a resource-limited wearable such as our CC2640R2F MCU (Section 3.2.3). Figure 3.11 and Figure 3.12 show performance of the classification model in detecting *eating* or *non-eating*, when the top k features were used ($5 \leq k \leq 60$).

3.4.5 Classification Aggregation

Given the classification results produced by the classifier on each 3-second window, we then decided to aggregate these results into coarser windows. We conducted a two-stage aggregation process. In Stage A, since completing a mouthful usually lasts longer than 3 seconds, we chose to aggregate prediction results of twenty 3-second time windows to a result every 1 minute according to a threshold: if more than 10% of the windows in a minute were labeled *eating*, we labeled that minute as *eating* (shown in Figure 3.13). The evaluation results in Section 3.5.1 are based on the results after Stage A aggregation. Additionally, in Stage B, we aggregated 1-minute prediction results from Stage A to *eating episodes*, which can last for several minutes. We used 50% overlap between consecutive 1-minute time windows. We used one parameter γ to achieve eating episodes: if the gap between two 1-minute time windows' prediction result was less than γ, we merged them into one eating episode (shown in Figure 3.14). We chose γ = 15 minutes, which is same as δ used in our definition of eating episode (Section 3.1). The evaluation results in Section 3.5.2 are based

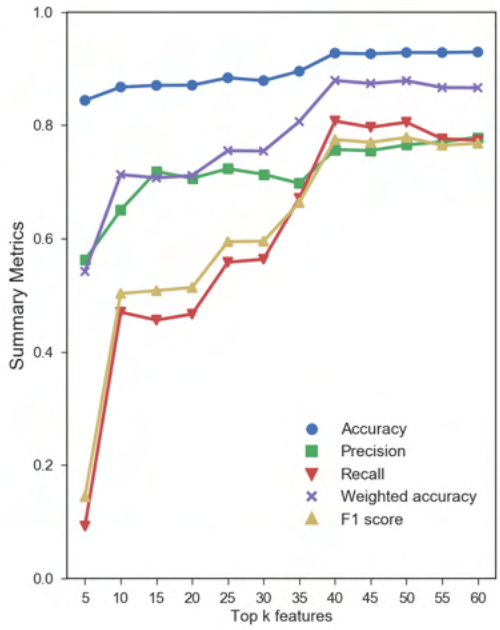

Figure 3.11: Results when using only field data for training classification model

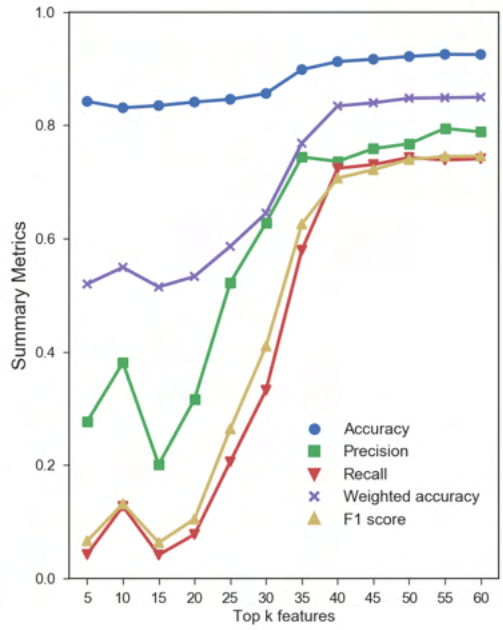

Figure 3.12: Results when using both field data and additional eating data for training classification model

on the results after stage B aggregation.

3.4.6 Ground-truth Label Aggregation

We used a similar two-stage aggregation approach on the ground-truth data to obtain ground-truth labels of 1-minute windows and eating episodes, and used them for window-based evaluation (Section 3.5.1) and episode-based evaluation (Section 3.5.2), respectively. In Stage A, we aggregated the ground-truth labels using the same method and threshold as in Section 3.4.5. In Stage B, we merged the 1-minute ground-truth labels into eating episodes using our definition in Section 3.1.

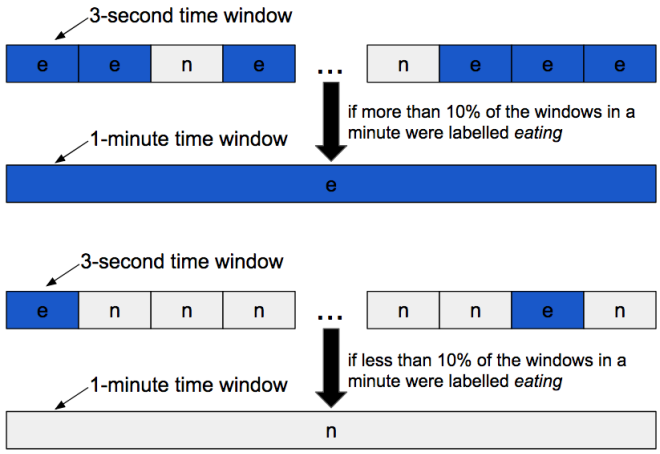

Figure 3.13: Stage A aggregation (e indicates time window labeled as eating; n indicates time window labeled as non-eating)

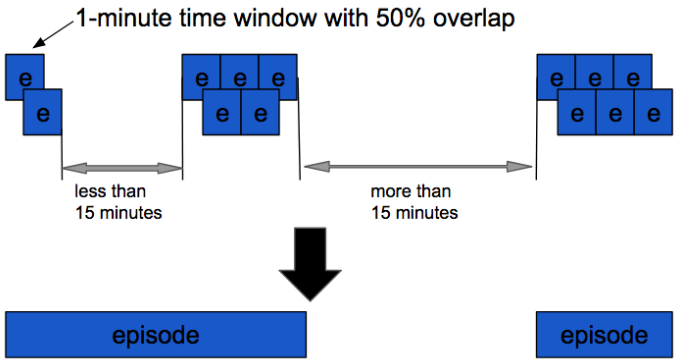

Figure 3.14: Stage B aggregation (e indicates time window labeled as eating; episode indicates eating episode)

3.5 Performance evaluation

To evaluate the performance of our approach, we evaluated Auracle's accuracy at two levels of detail: how well Auracle detected short periods of eating (using 1-minute windows of data) and how well those windows were aggregated into longer eating episodes.

3.5.1 Window-based Evaluation

Using the LOPO cross validation from Section 3.4.1, Figure 3.11 shows how well our classifier detects eating and non-eating data windows, when we vary the number of top

features, k, from 5 to 60. In the experiment, adding features improved the F1 score up to $k = 40$, after which adding more features yielded little-to-no improvement. To achieve a reasonably high F1 score and avoid high power consumption when we later run feature-extraction algorithms in a wearable platform, we chose to use the top 40 features (Table 3.1) for evaluation in Section 3.5.2 and implemented these features in the MCU of our prototype (Section 3.2.3).

We also tried adding the laboratory-based eating data we collected in Section 3.3.2 into the training data set for each iteration of LOPO cross validation, and explored whether it helped to improve results. Figure 3.12 shows the performance of the classification model for different feature set sizes. Table 3.3 shows summary metrics in the two above cases when using top 40 features. From the figure and table, we see that the addition of this data did not improve the classification performance. We speculate that the reason is eating behaviour of participants in the laboratory and free-living conditions are different. Participants sat and ate without many body movements in the laboratory, but they sometimes ate while moving (and even walking) in free-living conditions.

To better understand the difference between the eating data collected in the laboratory and free-living conditions, we conducted another experiment. We trained another LR classifier with all the field data and used all the laboratory eating data for testing. The data prepossessing and feature extraction and selection approach are same as those mentioned in Sections 3.4.2 and 3.4.3. We found this classifier could only recognize 61.9% of laboratory eating data as *eating* and misclassified other laboratory eating data as *non-eating*. As a result, adding eating data collected in the laboratory setting did not help the classifier to better recognize eating in free-living conditions.

3.5.2 Episode-based Evaluations

According to our definition of *eating episode* in Section 3.1, there were 26 actual eating episodes in our field data, ranging in duration from 1 minute to 41 minutes. As shown in

Table 3.3: Results when using field data only and combining additional eating data for training (mean value ± standard deviation)

Training data	Accuracy	Precision	Recall	Weighted accuracy	F1 score
Field data	0.928 ±0.042	0.757 ± 0.158	0.808 ±0.133	0.879 ±0.074	0.775 ±0.128
Field data with additional eating data	0.913 ±0.047	0.736 ± 0.155	0.724 ±0.224	0.834 ±0.108	0.707 ±0.174

Table 3.4: Results for episode-based evaluation using Jaccard similarity coefficient

	Ground truth	CD	MD	FD
Number	26	20	6	12
Maximum duration (minutes)	42	49	42	21.5
Minimum duration (minutes)	1.5	1.5	0	1
Mean duration (minutes) ± standard deviation (minutes)	17.8 ±11.9	19.7 ± 10.9	12.8 ±15.8	5.2 ±6.7

Table 3.4, when using the Jaccard similarity coefficient, we correctly detected 20 eating episodes out of 26 and missed 6 eating episodes. We also falsely detected 12 eating episodes. For the Correct Detection (CD) cases, the mean and standard deviation of delay and duration difference were 3.0 ± 3.8 minutes and 5.3 ± 5.9 minutes. Because we aggregated 1-minute time windows with 50% overlap to eating episodes, the resolution of episode-based evaluation is 30 seconds. In other words, our method will take at least 30 seconds to detect an eating episode.

To understand the source of Missed Detection (MD), we visually analyzed the data. In certain cases we identified the eating episode correctly, but within the subsequent (or previous) 15 minutes, the participant performed an activity (e.g., face touching) that our 1-minute window inferred as eating. This widened the span of the detected eating episode, with low overlap between the detected eating episode and actual eating episode. Thus, the Jaccard similarity coefficient in these scenarios was less than 55% and the eating episode was considered as MD. Figure 3.15 shows an example.

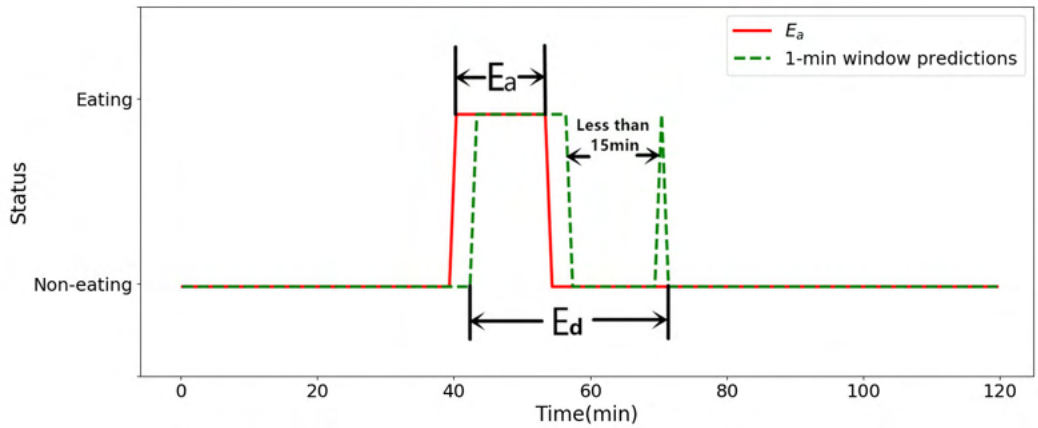

Figure 3.15: An example of Missed Detection (E_a indicates the actual eating episode; E_d indicates the detected eating episode)

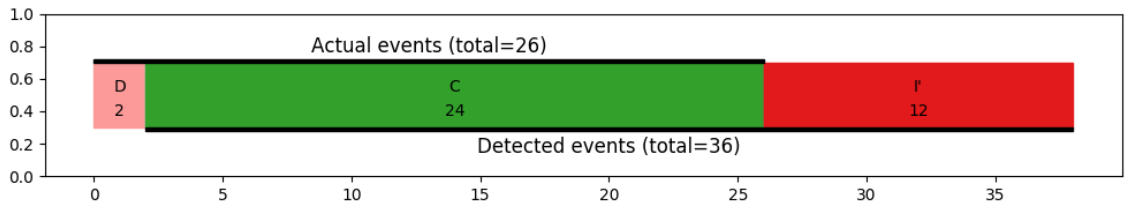

Figure 3.16: Results for episode-based evaluation using Ward's metrics

In addition, we evaluated our method to detect *eating episodes* using Ward's metrics [80]. As shown in Figure 3.16, we achieved 24 correct detection (C) among 26 actual eating episodes with 12 false insertions (I') and 2 deletions (D). Figure 3.17 shows the *eating episode* assignment for 14 two-hour sessions in the field study.

3.6 Power and memory evaluation

In this section, we estimate the power consumption of the Auracle during operation. Although the current prototype runs continuously at full power, we anticipate adding a wake-up circuit that would allow the MCU to remain in a lower-power 'sleep' mode when no sound is detected.

Figure 3.18 shows the wake-up circuit, which detects a surrogate measure of signal

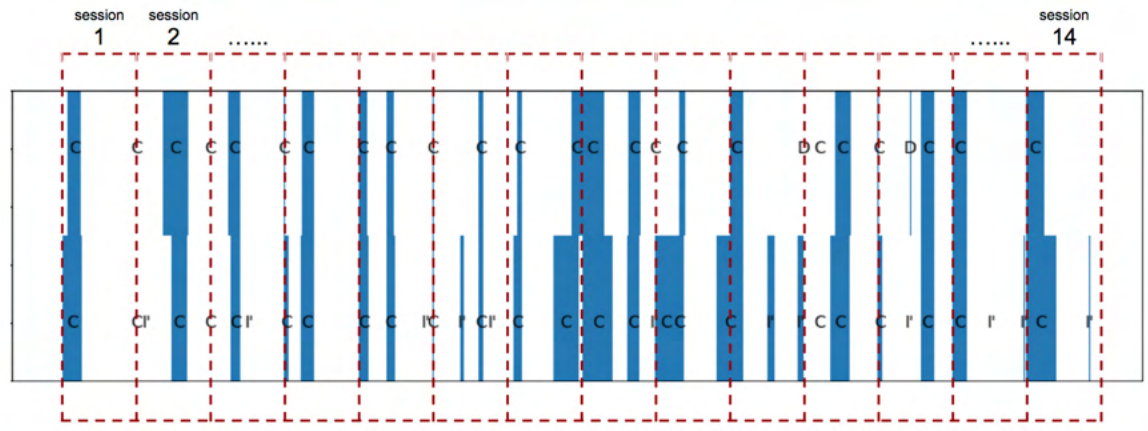

Figure 3.17: Eating-episode assignment for 14 field-study sessions (each red box indicates a different session)

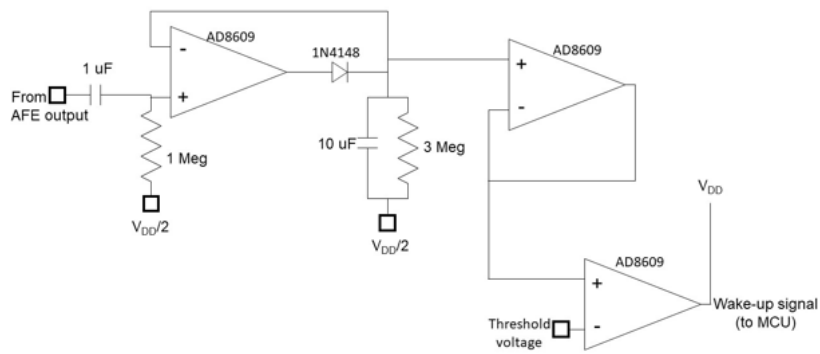

Figure 3.18: Wake-up circuit

variance and compares with a preset threshold. As shown in Figure 3.19, when the wake-up circuit detects sound, it triggers the MCU to switch from sleep state to wake-up state and begin sampling, processing, and recording data. This process is similar to the first stage of our classification model (Section 3.4.4), in effect replacing the first software stage with hardware and allowing the Auracle to stay in low-power sleep state more than half of the time. There are three AD8609 in the circuit and the V_{dd} is $3.3\,\mathrm{V}$. According to the data sheet, the total power consumption would be $0.5\,\mathrm{mW}$, which we used as the estimated power consumption of the wake-up circuit.

We model the power consumption of the Auracle, with that addition, but must first measure the consumption of the current prototype. We used a Monsoon Power Monitor

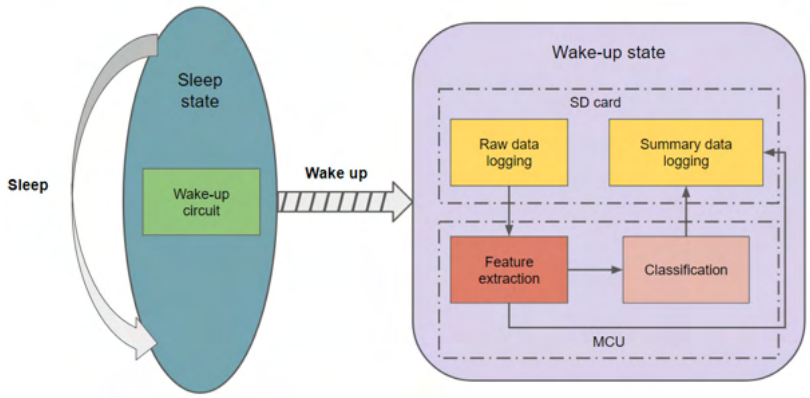

Figure 3.19: State diagram

(Monsoon Solutions Inc., FTA22J) to conduct all power measurements. For each measurement, we use the Monsoon to recorded power data for half a hour at each activity level, from which we calculated the average power consumption.

We first define three different modes: verbose mode P_v, development mode P_d, and realistic mode P_r. In *verbose mode*, the MCU logs both raw data and summary data to the SD card whenever it is not sleeping. In *development mode*, the MCU logs summary data to the SD card whenever it is not sleeping. In *realistic mode*, the MCU continuously transmits prediction results through BLE to a smart phone (and logs no data to SD card).

We estimated the power consumption in verbose mode P_v, development mode P_d, and realistic mode P_r as follows:

$$P_v = S * P_s + (1 - S) * (P_d + P_{c1})$$

$$P_d = S * P_s + (1 - S) * (P_d + P_{c2})$$

$$P_r = S * P_s + (1 - S) * P_d + P_b$$

where S is the fraction of time spent sleeping, and P_s indicates the power consumption when the system is sleeping. We estimated P_s by summing the power consumption of the wake-up circuit ($0.5\,\mathrm{mW}$, the power consumption of the contact microphone ($0.33\,\mathrm{mW}$, as shown in Section 3.2.1) and the power consumption of the MCU in standby mode ($0.06\,\mathrm{mW}$, as measured by Monsoon). By summing the power consumption of these three parts, we achieved P_s to be $0.89\,\mathrm{mW}$. Based on the fraction of 3-second windows when the audio

Table 3.5: Average power consumption of each component

	Average Power Draw (mW)
Sleep state (P_s)	0.89
Data processing (P_d)	+18.29
Summary data logging (P_{c1})	+2.29
Raw data and summary data logging (P_{c2})	+7.28
BLE (P_b)	+3.37

signal was below threshold in the field data we collected (Section 3.3.1), we estimate that S = 0.503.

P_d indicates the power consumption when the MCU samples sensor data, and runs feature extraction and classification algorithms on chip. We achieved P_d by directly measuring the power consumption of our PCB when data is processing on board.

P_{c1} indicates the power required to write the raw data ($500\,\mathrm{Hz}$ sampling rate, 1000 bytes written per second) and summary data (feature values and prediction results; less than 200 bytes written per 3 seconds) to the SD card. P_{c2} indicates the power required to log only the summary data to SD card. We determined both P_{c1} and P_{c2} by calculating the difference in the power consumption of our PCB with and without SD-card writing enabled.

P_b indicates BLE power consumption of our PCB when transmitting only classification results (2 bytes per 3 seconds) to an iPhone via BLE. We used the TI BLE-Stack software development kit to interface with the on-chip BLE radio, and the LightBlue[3] iOS and Android app to receive the data on smartphone. We determined P_b by calculating the difference in the power consumption of our PCB with and without BLE transmission enabled.

Based on the assumptions above, we estimated the power consumption of each component in Table 3.5 and power consumption in each system mode in Table 3.6.

Auracle is powered by 3.3 V. Assuming use of a 110 mAh battery, we estimated Auracle

[3]https://punchthrough.com/

Table 3.6: Average power consumption in each system mode

System Mode	Average Power Draw (mW)
Verbose mode (P_v)	13.16
Development mode (P_d)	10.68
Realistic mode (P_r)	12.91

can last 27.6 hours, 34.0 hours, and 28.1 hours in *verbose mode*, *development mode*, and *realistic mode*, respectively.

We also implemented our feature extraction and classification algorithms in the 20 KB SRAM of the MCU. Based on our measurement of the memory usage, we used 8.2KB SRAM when the MCU is in sleep date, and 19.2KB SRAM during other periods.

3.7 Discussion

Handling misclassification: To identify the reasons that lead to misclassification of eating as non-eating and vice versa, we watched the videos during all the periods that were misclassified by our system. Some scenarios where false positives occurred include instances when the monitored individual was talking while walking, continuously touching their face, excessively moving their body, or making constant contact between neck or hoods and the mechanical housing. We also observed that several false negatives occurred when the individual was eating while walking or eating a soft food item like yogurt. We found that among all these reasons, the motion artifacts caused by walking and body movement played an important role in the misclassification. We believe that adding an accelerometer or an IMU to Auracle may reduce the effect of the motion artifacts. Another possible technique to reduce classification errors is to design non-standard features based on the data. In addition, a number of classifier fusion methods (e.g., fuzzy templates) could lead to potential improvement in the classification performance [59].

Additional sensing modality: Auracle relies on chewing detection. Based on the chewing

action, Auracle determines whether a person is eating. However, if a participant performed an activity with a significant amount of chewing but no swallowing (e.g., chewing gum), which is not 'eating' based on our definition, our system may output false positives. Fusing data from additional sensors (e.g., a throat microphone for swallowing detection or wrist-worn devices for eating gesture recognition) might help handle situations that involve chewing but are not eating.

Mechanical design: The current design of Auracle works for individuals with standard head-shapes and is compatible with eyeglasses. However, we noticed that the standard deviation of F1 score for eating detection results among all the participants is relatively large (0.128 as shown in Table 3.3). One reason could be that the pressure between contact mic and the skin of some participants was significantly different from that of others. More specifically, the mechanical housing was either too tight or to loose for them. More personalization of mechanical design can be explored to ensure Auracle can fit better for different head shapes.

Address the need of health-science researchers: Auracle is a device designed for use primarily by health-science researchers. With better understanding about the needs of researchers, we can further improve our system. For instance, what accuracy and resolution of eating detection do researchers really need when under various research goals? The answer to this question will clarify the direction for fine tuning our device in the future.

3.8 Related work

Health-science researchers are interested in various measurable parameters including eating-specific data such as the time, duration and rate of eating, and meal-specific data such as food quantity, food group classification, and calorie estimation [52]. For all of these parameters, accurate recognition of *when* people eat is the foundation of effective automatic dietary monitoring (ADM) systems. Several review papers [2, 31, 52, 78] covered aspects of eating detection and summarized ADM systems developed. Here we focus only on technologies

developed to recognize when people eat in free-living scenarios.

Bedri et al. evaluated optical, inertial and acoustic sensors, and ended up using a behind-the-ear inertial sensor and achieved an F1 score of 80.1% for detecting eating episodes [5]. Using a proximity sensor, Chun et al. developed a necklace that captures head and jawbone movement [19]. They achieved 78.2% precision and 72.5% recall for detecting eating episodes in the free-living study. In another ADM system, Outer Ear Interface (OEI), three proximity sensors are encapsulated in an earpiece to monitor jaw movement by measuring ear-canal deformation during chewing [6,8]. In a field experiment, OEI classified five-minute segments of time as eating or non-eating with 93% (user dependent) and 82% accuracy (user independent) [8]. Thomaz et al. collected wrist-mounted audio data and tried to use ambient sound to infer eating activities [74]. Their system was able to identify meal eating with an F1 score of 79.8% in a person-dependent evaluation. Sen et al. built and tested an approach based on wrist motion and achieved false-positive and false-negative rates of 6.5% and 3.3% respectively [64, 65]. Zhang et al. evaluated smart eyeglasses they proposed in free-living scenarios and achieved precision and recall more than 77% for chewing detection [88]. Mirtchouk et al. experimented with different combinations of motion (head, wrist) and audio (air microphone) data collected in laboratory and free-living conditions [42]. They found a combination of sensing modalities (audio, motion) was needed; yet sensor placement (head vs. wrist) was not critical.

In these previous field studies, researchers logged field data in free-living scenarios and ran offline experiments. Even though we currently run experiments offline, the Auracle can do real-time eating detection. We developed an ADM system that can locally capture, process, and classify sensor data collected in out-of-lab, day-long, free-living scenarios.

3.9 Summary

In this chapter, we describe Auracle, a wearable system for eating detection in free-living scenarios. We first implemented the Auracle hardware, which includes a contact microphone, battery, wearable mechanical housing and PCB with data acquisition function. Using this device, we collected field data with 14 participants for 32 hours in free-living scenarios and additional eating data with 10 participants for 2 hours in laboratory scenarios, respectively. Based on these data, we designed a data-processing pipeline and evaluated its performance using LOPO cross validation. We achieved accuracy exceeding 92.8% and F1 score exceeding 77.5% of eating detection, and successfully detected 20-24 eating episodes (depending on the metrics) out of 26 in free-living conditions. Finally, we implemented the data-processing method on our prototype and estimated the power consumption of Auracle. We anticipate Auracle can last 28.1 hours with a 110 mAh battery in realistic mode.

4

Adapting the approach for children

As noted in the introduction, obesity has become a serious threat to public health in America. In most cases, obesity is caused in part by over-consumption of food, so individualized feedback about eating habits may help reduce obesity rates. This information is most pertinent early in the lifespan, prior to excess weight gain and the development of obesity. Childhood obesity rates continue to be high (18.5 percent in 2016) in the United States [28] and are associated with a myriad of co-morbidities that negatively impact overall quality of life [63]. Furthermore, weight-related issues in childhood are likely to carry into adulthood [81]. It is therefore essential to improve our scientific understanding of childhood eating behaviors to inform obesity interventions. Indeed, individualized, just-in-time adaptive interventions

(JITAIs) focused on eating habits may be effective in reducing over-consumption in children [34], but are not feasible until there is technology that can automatically detect and measure eating behavior.

To monitor eating behavior in children, we face all those challenges we mentioned in Chapter 3 and more: children usually have more non-eating related head and body movement during eating, children have more complex eating behaviour (e.g., children may hold and play with food in their mouths for a while before chewing and swallowing), children's head and body sizes vary more than adults, and children are more sensitive to the discomfort of wearable devices [44, 84]. Although several researchers have evaluated ADM systems on adults, no automatic dietary monitoring technique exists for children. Researchers and behavioral scientists depend on traditional techniques such as video coding and manual food journals to monitor dietary activities among children [51]. To better support the needs of clinicians and behavioral scientists in monitoring eating habits among children, we modified the Auracle system, which had previously been tested only among adults.

In this chapter, we report the insights gained and results obtained from experiments with a new child-oriented ADM system derived from the Auracle. To evaluate its performance, we conducted a set of controlled experiments. During this study, the participants (children) visited our laboratory on multiple occasions and consumed a variety of meals while wearing the modified Auracle system. Our initial findings indicate that it is indeed possible to identify and monitor fine-grained eating activities of children, once we addressed specific challenges. With further refinement, we believe that such an ADM system may also be used to monitor a child's eating activity in naturalistic settings.

Accurate high-resolution eating detection could help trigger other kinds of sensing or inquiries [11]. Specifically, we believe it is important to develop ADM techniques that can detect eating (whether the user is eating or not), within a few seconds of eating onset, to enable (1) detailed analysis of eating patterns like mouthful rate, chewing rate, and consumption rate, and (2) to enable just-in-time interventions in free-living conditions.

For instance, researchers have recently shown that poor mastication is associated with obesity [71]. Additionally, if we want to estimate the caloric intake of a meal, we may need to classify different types of food consumed during the meal, and thus require eating detection to identify the precise moment of mastication for each food item. We set out to enable such capabilities for monitoring children, and believe it to be the first effort to do so.

In the work described in this chapter, my contributions include the following.

- I coordinated the adaptation of Auracle hardware and mechanical design to allow data collection from children.
- I designed a study in both meal and snack scenarios involving 10 children over a total of 60 lab sessions.
- I improved the feature-extraction stage in our data-analysis approach to achieve better performance.
- We achieved an accuracy exceeding 85.0% and an F1 score exceeding 84.2% for eating detection with a 3-second resolution. The same methods obtained a 95.5% accuracy and a 95.7% F1 score for eating detection with a 1-minute resolution.

Regarding the work described in this chapter, I acknowledge the contributions of others:

- Yiyang Lu assisted with the data collection and implementation of the data-analysis approach.
- Nicole Tobias designed the updated Auracle's PCB.
- Ella Ryan developed the elastic headband in the revised Auracle prototype.

For these above-mentioned contributions, I provided collaborative input.

4.1 Background

Researchers have developed ADM systems that use various cues for eating detection, including audio collected from the ear canal [1,39,47,49,61,67,77], behind the ear [11,12,89] or on the throat [46, 55, 61, 62, 85], proximity of the necklace from the chin [19], first-person

images from chest-mounted cameras [56, 69, 73], or wrist-based gesture recognition [22, 64]. Each approach has been tested on adults and has its own limitations and advantages. We think a suitable device for children should avoid or minimize following factors: danger in free-living conditions (e.g., tiny microphone in the ear canal), privacy violation (e.g., images capturing the child, or other children), social awkwardness (e.g., device on throat), or distraction during regular use (e.g., a wristband on the dominant hand or on both hands). Furthermore, any ADM system aimed at free-living conditions must be accurate, compact, light, comfortable, cheap, robust, usable, and energy efficient.

Rather than starting from scratch, we sought to adapt an adult ADM system, one with most of those properties and a comprehensive evaluation of its accuracy and power consumption. As shown in Chapter 3, Auracle demonstrated success on an adult population, in both lab and free-living settings.

Additionally, Auracle is a head-mounted device with a form factor similar to a behind-the-head pair of earphones (Figure 3.4); we believe a professionally-engineered version of this design would be smaller, safe, and comfortable for a child to wear. This design places a skin-contact microphone behind the wearer's ear, to capture the sound of a person chewing; this approach should be safer than placing a microphone or other sensor in the ear canal, and less disruptive to normal hearing. Since the Auracle is out of view of the child, we speculate that it might be less distracting than anything worn on the top or front of the head. Nonetheless, for our work in this chapter we developed a new approach (details in Section 4.2) that we believe is an even more natural choice.

4.2 System design

The Auracle system includes a contact microphone, a battery, a custom-designed PCB for data acquisition and a wearable mechanical housing (Figure 3.4). Since the device was primarily designed for data collection with adults, we had to modify the housing of the

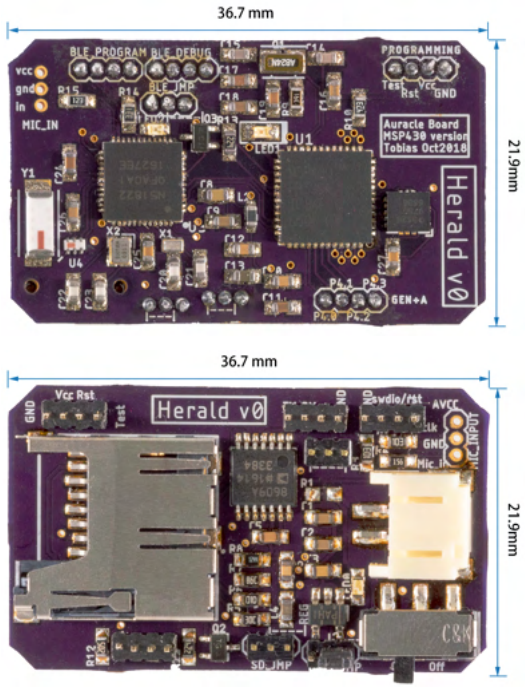

Figure 4.1: The top and bottom view of the updated and improved Auracle's PCB.

device to ensure that it performed reliably in detecting children's eating activities while ensuring that it did not distract or discomfort the child.

For this study, we updated both the hardware and software of Auracle. The updated PCB had several new or improved features relative to the version described in Chapter 3. These updates included replacement of the original Texas Instruments (TI) CC2640R2F MCU with the MSP430FR5994 MCU, addition of a new BLE chipset (Nordic nRF51822), and addition of an accelerometer (ADXL362). We did not use the accelerometer or BLE communication in our current study. We used the Auracle to collect 10-bit samples of the microphone signal at 500 Hz and write the data to the SD card. In this study, the most beneficial aspect of the updated Auracle hardware was that the total size of the board was reduced by over 50% (Figure 4.1): now smaller than 37×22mm; it was easier to use this board to design a device suitable for the smaller heads of children.

Based on our preliminary tests, we observed that children's heads vary tremendously in size and shape, making it necessary to design a form factor that could easily adapt to

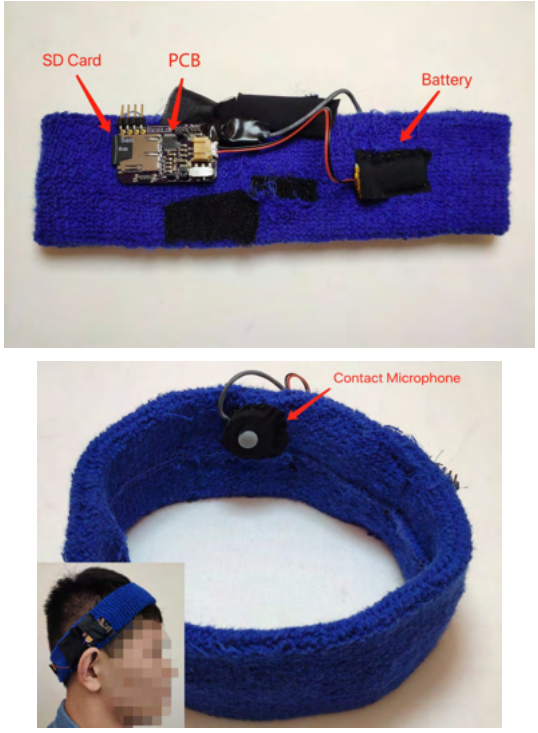

Figure 4.2: Auracle prototype after our revision, using elastic headband.

a range of children. Although the Auracle microphone's position and the pressure that it applied to the skin were adjustable in the original design, preliminary testing showed that it did not provide adequate contact for several children, rendering the collected data inadequate for analysis. This observation prompted us to house the Auracle in an elastic headband (Figure 4.2) rather than in the original 3d-printed plastic frame (Figure 3.4). The elastic headband ensured that the device was comfortable and robust to movement, and the microphone maintained proper contact with the child's skin. It also adapted to a wide range in head sizes without requiring any mechanical design modification. Furthermore, it was less distracting for the child during the in-lab studies.

4.3 Data collection

To determine the usefulness of ADM systems in a health-science study, we partnered with a research group that studies eating behavior in children. We trained several research assistants

to use our modified Auracle in their study, following a protocol approved by our IRB.

4.3.1 Laboratory data collection

We collected data from 10 children (aged 4-17; 4 female, 6 male). Each participant visited the lab on three occasions and we collected data from two sessions per visit: one meal and one snack. Overall, our dataset consists of 30 meal and 30 snack sessions. After a preliminary review of the data, we determined that we could not use data from 16 sessions, collected from 4 different participants, for the following reasons. In four of these sessions, the contact microphone signal was weak due to poor contact or improper placement of the microphone, and the signal barely changed during these sessions. In the other twelve sessions, the data was not usable because research assistants forgot or incorrectly performed some of the procedures in our protocol (e.g., turn on the camera, start with three-tap event) or because participants inadvertently interfered with the Auracle (e.g., touching the headband frequently). For our final analysis, we excluded the data collected during these 16 sessions. We used the remaining 44 sessions of recorded data (16.86 hours in total) from 8 participants for further analysis.

4.3.2 Data collection protocol

At the start of each session, a research assistant placed the Auracle device around the participant's head and adjusted the contact microphone so that it was located on the mastoid tip, behind the ear. The participants were instructed not to adjust or remove the Auracle device during the study. We placed a Go Pro camera in front of the participants to film their eating behavior. We later annotated the videos to provide 'ground truth' about the participants' eating behavior.

We first conducted the 'meal' session, in which we served pre-determined portions of three food items to participants (macaroni and cheese, carrots, and apple slices). Participants sat in front of a dining table during the meal, and we encouraged them to perform the eating

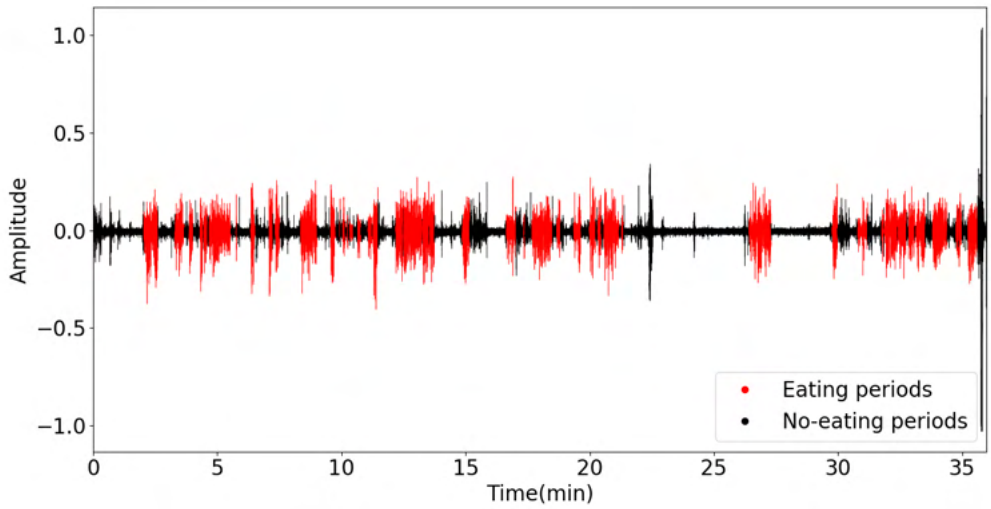

Figure 4.3: Temporal signature of one session of data collection (red portions indicate periods of eating).

activity as they normally would in a naturalistic setting.

Participants were provided additional servings of any food type if they completed the initial serving and indicated that they wanted more of that food type. After a short break, the 'snack' session began: we provided another three food types (gummy bears, grapes, and goldfish crackers) to the participants. Participants sat on a sofa, in front of a TV, watching a show (with commercials about food) for 30 minutes.

A example of one session of data collection is shown in Figure 4.3. The red portions in the figure were human-annotated to indicate eating periods. Figure 4.4 shows two screen shots of the video recorded by the camera from two participants during the meal and snack sessions, respectively. In general, we found that participants were more relaxed and natural in the snack session than the meal session. Overall, none of the participants complained about any discomfort caused by the device and did not remove it during their sessions.

At the beginning of each session, we asked the participant to simultaneously tap on their cheek and on the headband three times using their hand, while ensuring that the camera could record this action. At the end of the session, we asked the participant to again perform this 'triple-tap' action. We later identified these triple-tap events in the video (from the

Figure 4.4: Screen shots of the video recorded during meal and snack sessions.

camera) and the audio (from the microphone) and used them to synchronize the video and audio data streams.

During the data collection, at least one research assistant was always present in the room with the child for safety reasons. The research assistant visually checked the position of the headband periodically to ensure the device stayed at the proper location during the study. However, the research assistant pretended to focus on paperwork, and avoided talking to or distracting the participants. We also asked one parent of the child to wait near the laboratory, to address any unexpected situations. We compensated each participant with a $30, $35, or $40 gift card for the first, second, and third visits, respectively.

4.3.3 Video annotation

We used a commercial service to annotate the videos.[1] The annotation process consists of three steps: execution, audit, and quality inspection. In the *execution* step, an annotator

[1]BasicFinder: https://www.basicfinder.com/en/

watched the video and annotated each period of eating, at a 1-second resolution. Thus, for every second in the video, the annotator indicated whether the child was eating or not. Next, in the *audit* step, an auditor watched the video and checked whether the annotations were consistent with the content in the video. The auditor noted any identified inconsistency for the next step: quality inspection. Finally, in the third step, a *quality inspector* reviewed the questionable labels and made the final decision about each identified inconsistency. The *quality inspector* also conducted a second-round inspection of 20% of the samples that were considered consistent during the previous two inspection rounds. This three-phase, three-person process ensured that the quality of the video annotation was acceptable.

4.4 Data analysis

We next describe our evaluation metrics, and the stages of our data-processing pipeline: preprocessing, feature extraction, classification, and aggregation.

4.4.1 Evaluation metrics

We set out to evaluate our method for fine-grained eating detection *(window-based classification)* and for coarse-grained eating detection *(episode-based classification)*, as detailed in the subsections below. Since we aim for generalized models, we use a Leave-One-Session-Out (LOSO) approach to evaluate model efficacy. In a LOSO approach, data from one session of a participant is tested on a model that has been trained using a combination of data from all other sessions of the same participant and every session of all other participants. Formally, if the dataset has data from I participants, each of whom has provided data for J sessions, then set S_{ij} represents participant i's data from session j, for $i \in \{1, 2, \cdots I\}$ and $j \in \{1, 2, \cdots J\}$. Overall, set $S = \cup_{\forall i,j} S_{ij}$ represents all sessions in the dataset. Then the model is trained using sessions in the set $S - S_{ij}$ and tested on session S_{ij}. This process is repeated so that every session of every participant is tested on a model generated from all

sessions in the dataset, except the session being tested.

In preliminary tests, we observed that the data we collected from a participant in different sessions could often vary in signal amplitude. One reason for this difference is because the same participant might wear the Auracle device differently (e.g., the angle of wearing the headband) during different sessions, which caused the contact microphone to be located at different locations or in contact with the skin with different pressure. Moreover, actions during the session (such as touching the device during the session or scratching the head) may also have affected the microphone contact. Thus, we first applied the normalization approach mentioned in Section 4.4.2, and then chose a LOSO cross-validation approach to test the performance of the classifier in detecting the eating activity for data in a session that it has never seen before.

LOSO window-based evaluation

In window-based evaluation, we explored two window sizes: 3 seconds and 1 minute. Three-second windows are important for applications that rely on the output of ADM systems to drive fine-grained interventions (e.g., an in-the-moment intervention based on the mastication habit). One-minute windows enable us to compare our results with results presented in Chapter 3. For each window size, we compare our classifier's output against the ground-truth label for the corresponding time window, computing four evaluation metrics (accuracy, precision, recall, and F1 score) for each session, then averaging those metrics across sessions to compute the four summary metrics for that window size. We used the same metrics as the evaluation in Chapter 3.

LOSO episode-based evaluation

In episode-based evaluation, using an approach similar to previous work by Papapanagioto et al. [48], we matched each detected eating episode with either 0 or 1 ground-truth eating episode. We used the Jaccard Similarity coefficient to determine whether this match led to

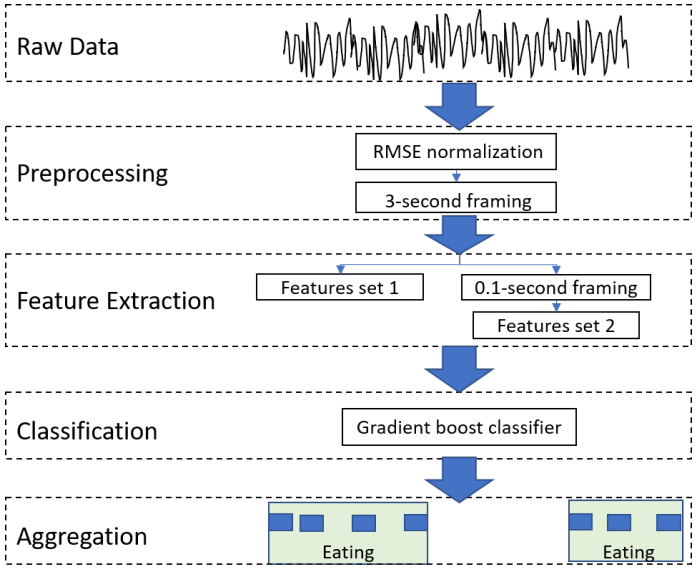

Figure 4.5: Data processing pipeline.

a *Correct Detection*, *False Detection*, or *Missed Detection*. Our definition of the Jaccard Similarity coefficient is the same as in Chapter 3.

4.4.2 Data processing pipeline

Figure 4.5 presents our overall data-processing pipeline, which comprises preprocessing, feature extraction, classification, and aggregation steps.

Preprocessing

The preprocessing stage includes three steps: root mean square energy (RMSE), normalization, and segmentation. As noted above, the audio-signal amplitude can be affected by the location of the contact microphone and the pressure applied to it. We observed that the signal amplitude varied from session to session due to differences in position, pressure, and head size/shape. To ensure uniformity, we used the root mean square energy (RMSE) value to normalize the signals within each session. However, we found motion artifacts in some portions of the signal; these artifacts were usually caused by movement of the contact microphone across the skin, and can have an outsized effect on the RMSE value.

After some preliminary tests, we decided to exclude samples that were not within the 95% confidence interval when we calculated the RMSE value of each session's signal. Then, in the normalization step, we divided all the data values by the RMSE value of the same signal. Then, we segmented the acoustic signals into non-overlapping 3-second windows of samples, and passed these windows to the feature-extraction stage. Note: our current normalization method computes RMSE across the entire session and is thus only suitable for offline processing; one can envision similar normalization approaches suitable for online processing, e.g., normalizing each sample by dividing by the RMSE computed over a period of recent data.

Feature extraction

For each time window, we extracted the 30 features shown in Table 4.1, including 20 frequency-domain features and 10 time-domain features selected from about 1,400 possible features (see Section 4.4.3 for details). We extracted some of these features directly from the windows received from the preprocessing stage, using the tsfresh[2] package. We extracted the other features using methods similar to those used by Bogdanov et al. [14], using the librosa[3] package. In this latter case, we segmented each 3-second window into 0.1-second 'frames' (with 75% overlap between adjacent frames) and extracted features from each frame. For each feature and each window, we obtained an array of values corresponding to the 0.1-second frames. We then computed eight statistics (mean, median, variance, maximum, minimum, kurtosis, skewness, entropy) for each array. Using these eight statistics of each feature array, we extracted features for each time window.

Classification and aggregation

The classification stage has two steps: classification and aggregation. First, the Gradient Boosting (GB) classifier used each window's features to classify that window into one

[2]v0.11.2: http://tsfresh.readthedocs.io/en/latest/
[3]v0.7.2: https://librosa.github.io/librosa

Table 4.1: 30 features used by our classifier (* indicates the frequency-domain features)

Feature category	Description	Number of features	Feature set
MFCCs*	Mel-frequency cepstral coefficients	14	2
MFCCs delta*	First derivatives of MFCCs	3	2
MFCCs delta 2*	Second derivatives of MFCCs	2	2
Spectral contrast*	Difference between the spectral peak and valley in each frequency subband	1	2
Change quantiles	Mean of the absolute change of the series inside a corridor	6	1
Agg autocorrelation	Value of an aggregation function over the autocorrelation for different lags	1	1
Agg linear trend	Attributes of a linear regression for values that were aggregated over chunks	1	1
Ratio beyond r sigma	Ratio of values that are more than r*std(values) away from the mean	1	1
Quantile	Value of the data point greater than q% of the ordered values	1	1

of two classes: eating or not eating. (See Section 4.4.3 for details about our selection of the GB classifier). Specifically, we used the GB implementation from XGBoost.[4] Using the same aggregation methods as in Chapter 3, the aggregation step combines groups of twenty 3-second windows' classification outputs into 1-minute outputs, and then further combines these 1-minute outputs into an episode-level output. We also apply the same two-step aggregation process to the ground-truth labels (which have a base resolution of 1 second).

4.4.3 Classifier and feature selection

At this point we digress to justify our choice of the GB classifier and our selection of the 30 specific features listed in Table 4.1. To make these decisions, we conducted several benchmark studies to determine the best-performing classifier (in terms of F1 score) and the

[4] v0.9.0: https://xgboost.readthedocs.io/en/latest/python

Table 4.2: Classifier performance when using top-30 features from only feature set 1

Classifier	Accuracy	Precision	Recall	F1 score
Gradient boosting	0.819	0.810	0.839	0.815
Random forest	0.816	0.815	0.820	0.809
K-nearest neighbors (K=5)	0.802	0.793	0.814	0.796
Logistic regression	0.793	0.818	0.776	0.786
Support Vector Machine	0.813	0.813	0.818	0.807
Gaussian Naive Bayes	0.802	0.760	0.884	0.809

most discriminative features.

Choice of classifier

We initially assumed we would use the Logistic Regression (LR) classifier because LR provided the highest F1 score in eating detection for our evaluation in Chapter 3. We decided to re-visit this selection, however, because we wanted to explore a broader range of features, and because that study was conducted with adult participants, consuming different food types, and in free-living conditions. It seemed plausible that a different classifier, and different feature set, would be better suited for eating detection in children, or in lab settings.

We began with a large set of 750 features extracted with tsfresh; let that be called *feature set 1*. After inspecting these features, we found many of them are constant numbers and not useful for classification. We also anticipated it was not necessary to use all the features, given the results in Chapter 2 and Chapter 3. For these two reasons, we decided to select the best classifier when using a smaller number of features. (More discussion about feature selection can be found in Section 4.4.3.) We then ran our entire dataset through our data pipeline, using only the top-30 features selected from feature set 1, using each of six common classifiers, resulting in the metrics shown in Table 4.2. (We found adding more features yielded little-to-no improvement to F1 scores, across these six classifiers.) In Figure 4.7 and Figure 4.8, we use the GB classifier as an example to show the performance of our model when top k features were used ($1 \leq k \leq 60$).

For further confirmation, we added a set of 650 features extracted with librosa (as

Table 4.3: Classifier performance when using top-30 features from both feature sets 1 and 2

Classifier	Accuracy	Precision	Recall	F1 score
Gradient boosting	0.850	0.834	0.869	0.842
Random forest	0.845	0.841	0.842	0.833
K-nearest neighbors (K=5)	0.823	0.805	0.847	0.818
Logistic regression	0.829	0.839	0.828	0.822
Support Vector Machine	0.840	0.832	0.851	0.832
Gaussian Naive Bayes	0.825	0.820	0.829	0.815

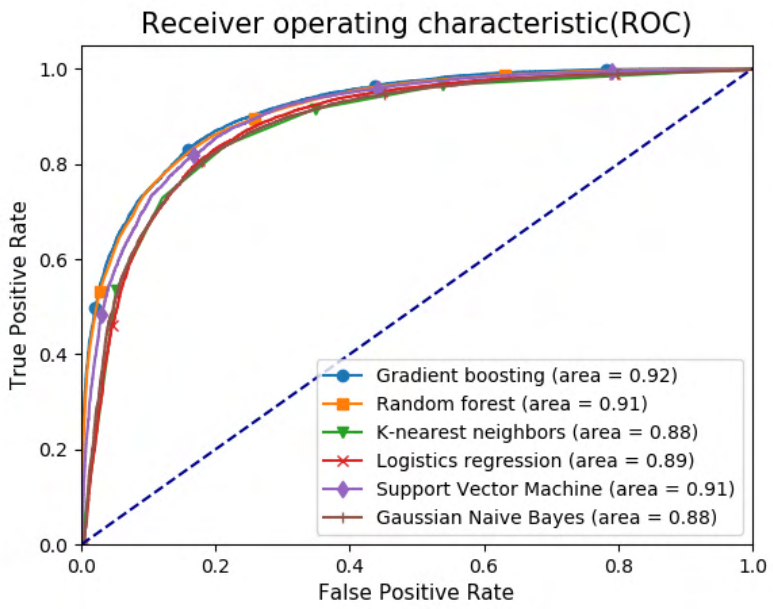

Figure 4.6: ROC curve for various classifiers.

described above); let that be called *feature set 2*. We again ran our entire dataset through our data pipeline, using the top-30 features selected from both feature set 1 and 2, using the same six classifiers, resulting in the metrics shown in Table 4.3. All six classifiers achieved a better F1 score relative to Table 4.2 (average improvement 2.3%, with a p-value of 0.0007), indicating that features in feature set 2 were indeed useful in the classification process.

Finally, for deeper insight into the differences among the classifiers, we plotted the Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC) in Figure 4.6 for all classifiers; it displays the relationship between the true-positive rate and the false-positive rate of our models.

Based on results in Table 4.2, Table 4.3 and Figure 4.6, GB, Random Forest, and Support Vector Machine outperformed the other three classifiers. Although the best three classifiers had similar performance, GB was slightly better so we selected GB for our analysis.

Feature selection

To determine how many features to use, and which features to use, we began by computing about 1,400 features (feature sets 1 and 2).

For feature set 1, we extracted 62 categories of common features directly from windows produced by the preprocessing stage; 4 frequency-domain categories and 58 time-domain categories. In each feature 'category', we extracted features with all possible parameters. Some feature 'categories' can result in hundreds of features by varying the category's parameters. In our case, feature set 1 consisted of about 750 features.

For feature set 2, we extracted 14 categories of frequency-domain features. Again, some feature categories can result in hundreds of features by varying the category's parameters. In our case, feature set 2 consisted of about 650 features.

Clearly, it would be too complex to compute all 1,400 features from these two feature sets, on small wearable platforms, so we used the Recursive Feature Elimination (RFE) algorithm to identify the subset of features that were most 'discriminative'. That is, we ran our entire pipeline (with each classifier) over the complete dataset, letting RFE empirically identify the subset of k features that were most useful in distinguishing eating from non-eating moments (for each classifier). As an example, Figure 4.7 shows the performance of the GB classifier in eating detection, when the top k features were used ($1 \leq k \leq 60$), with a 3-second resolution. From the figure we can see that the performance improved until $k = 30$ and then it saturated. When experimenting with other classifiers, we found the trend of curves are similar.

To further understand the effect of k, using the aggregation method mentioned in Section 4.4.2, we computed the performance of the system at a 1-minute resolution. Figure 4.8

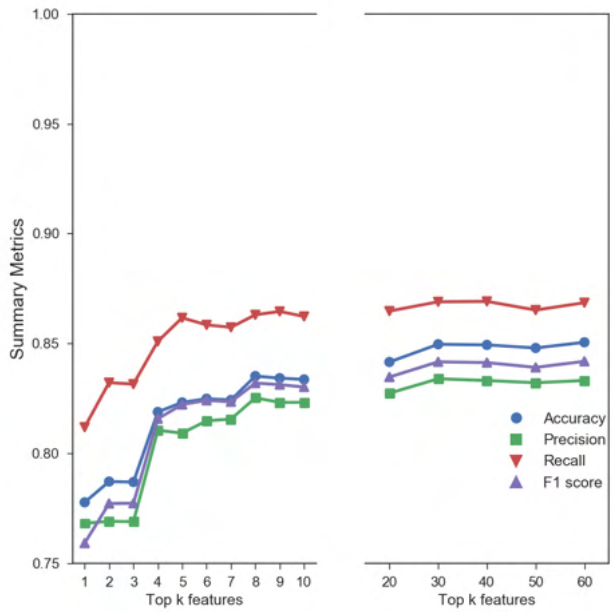

Figure 4.7: Performance of the GB classifier with 3-second resolution.

shows the performance of the GB classifier in eating detection, when the top k features were used ($1 \leq k \leq 60$), with a 1-minute resolution. Interestingly, saturation at the 1-minute resolution occurs even at lower k, indicating that the system can perform adequately with a small feature set. As previously mentioned, one of our long-term goals is to provide interventions based on fine-grained eating-related actions, so we decided to use $k = 30$ and list the top-30 features in Table 4.1. As it happens, 20 features out of 30 are frequency-domain features. Most of the frequency-domain features are Mel-frequency cepstral coefficients (MFCCs), and the first and second derivatives of MFCCs that were obtained from feature set 2, thus showing the usefulness of the features from the librosa package.

4.5 Performance evaluation

Overall, we evaluate how well our method worked

- for fine-grained eating detection (3-second windows),

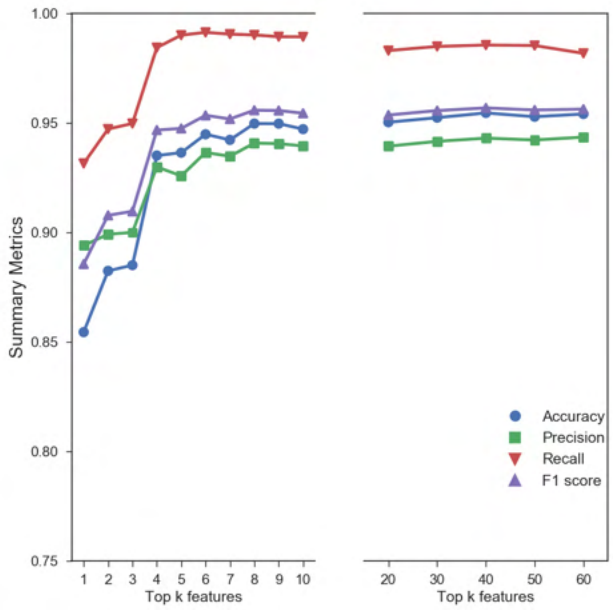

Figure 4.8: Performance of the GB classifier with 1-minute resolution.

- for medium-grained eating detection (1-minute windows), and
- for detecting eating episodes.

As noted above, we evaluated our method using LOSO cross validation; Table 4.4 summarizes the resulting performance metrics for fine-grained eating detection (3-second windows) and for medium-grained eating detection (1-minute windows). In these experiments we used the GB classifier and the top-30 features (Table 4.1), achieving an F1 score of 0.842 for fine-grained eating detection and 0.957 for medium-grained eating detection.

To better compare with our previous work in Chapter 3, we also performed an episode-based evaluation. According to our definition of *eating episode*, there were 45 actual eating episodes in our laboratory data. The episodes ranged in duration from 1 minute to 38 minutes, with mean value 20.01 minutes and standard deviation 12.09 minutes. When using the Jaccard similarity coefficient as our evaluation metric, and the aggregation methods mentioned in Section 4.4.2, we correctly detected 43 eating episodes, missed just 2 eating episodes, and falsely detected 0 eating episodes. We next examined the difference in duration

Table 4.4: Metrics for GB classifier with top-30 features (mean value ± standard deviation)

Training data	Session(s)	Accuracy	Precision	Recall	F1 score
3-second time window	meal	0.880 ±0.079	0.897 ± 0.094	0.923 ±0.050	0.907 ±0.063
	snack	0.820 ±0.085	0.771 ± 0.200	0.815 ±0.139	0.776 ±0.165
	both	0.850 ±0.088	0.834 ± 0.167	0.869 ±0.120	0.842 ±0.140
1-minute time window	meal	0.991 ±0.032	0.990 ± 0.037	1.000 ±0.000	0.995 ±0.020
	snack	0.918 ±0.124	0.896 ± 0.208	0.971 ±0.064	0.919 ±0.167
	both	0.955 ±0.097	0.943 ± 0.155	0.986 ±0.047	0.957 ±0.124

between detected eating episodes and actual eating episodes; the mean difference was only 0.76 ± 3.56 minutes. To further challenge our model, we increased the Jaccard similarity coefficient starting from 0.55 and found we were still able to achieve the same performance when the we increased the coefficient up to 0.76.

Although our method seems highly effective at detecting eating episodes, we must note that our in-lab sessions were relatively short (the longest sessions is 40.35 minutes) and participants were eating more than half of the time in typical sessions (the time-length ratio of data labeled as *non-eating* and *eating* is 0.91:1). The situation would not be challenging for any episode detector, so we cannot draw any firm conclusions about our method's ability in this regard.

4.6 Related work

Eating behavior of children has long been a important topic in health-science research. Researchers have studied various issues related to *when* and *how long* children eat. For instance, Klesges et al. found that time spent eating in a meal correlates to weight, but not to the total meal time (i.e., time spent at the table) [33]. In studies pertaining to monitoring

children's eating habits, researchers depend on traditional techniques such as video coding and manual bookkeeping for recognizing when and how long children eat. By developing a wearable sensor that can accurately detect eating, we believe most of these studies could be completed with finer granularity, higher accuracy, and substantially less labor.

Another common way to assess children's eating behavior is through the evaluation of the eating-behavior micro structure, including aspects such as bite size, eating rate (bites/minute) and meal length. For instance, Llewwllyn et al. have shown that children with higher eating rate tend to have higher body weight [40]. Accurate recognition of *when* children eat is the foundation of ADM systems that help collect micro-structure eating information for health-science researchers. A wearable system usable in free-living settings could capture metrics about eating outside mealtimes, which can influence a child's health but would be difficult to measure with traditional methods.

4.7 Summary

In this chapter, we adapted Auracle and applied it in a study with children. Indeed, we believe this chapter represents the first work to develop and evaluate a wearable ADM system for children. Using our adapted Auracle device, we collected data with 10 participants for 60 sessions (22.3 hours) in meal and snack scenarios. We designed a data-processing pipeline and evaluated its performance using LOSO cross validation. Overall, we achieved an accuracy 85.0% and an F1 score 84.2% for eating detection with a 3-second resolution, and a 95.5% accuracy and a 95.7% F1 score for eating detection with a 1-minute resolution.

5

Computer-vision based approach

In this chapter, we present a computer-vision based approach to detect eating. Specially, our goal is to develop a wearable system that is effective and robust enough to automatically detect *when* people eat, and for *how long*, in free-living conditions.

CNNs have been established as a powerful method for image recognition and action recognition in videos [16, 32, 68, 75, 76]. Encouraged by these results, we applied CNN to eating detection using vision-based approaches. We used a miniature head-mounted camera for data collection and then (offline) trained CNN models for eating detection using images and videos, respectively. The camera is fixed under the brim of a cap, pointing to the mouth of participants (as shown in Figure 3.6) and continuously recording video (but not

audio) throughout their normal daily activity. We developed such a camera for collecting ground truth in our previous work (see Section 3.3.1); we now use that video itself as a more accurate (and more comfortable) way to detect eating.

Additionally, in recent years, similar systems have been widely used to collect ground truth for the field studies with ADM [4, 5, 91] systems. In the future, researchers may be able to run the ground-truth videos they collected using our proposed approach and compare the performance of their approach with ours, if they use similar methods for ground-truth collection. This opportunity could address one of the major challenges in the field of ADM – the lack of comparison between different approaches [9]. Furthermore, our approach could be used to assist in video annotation and thus reduce the video-annotation burden in the field of ADM.

In the work described in this chapter, my contributions include the following.

- I refined the cap-mounted camera we used for ground-truth collection (see Section 3.3.1) and made it suitable for video-collection in our proposed field studies.
- I conducted field studies with 10 participants and collected about 55 hours of video data for data analysis.
- I developed and evaluated four CNN models to detect eating: 2D CNN (with frame), 2D CNN (with optical flow), 3D CNN (with video), SlowFast (with video).
- I validated the feasibility for deploying a 3D CNN model in mobile or wearable platforms, when considering computation, memory, and power constraints.

Unlike the work in previous chapters, I worked independently on this project under the guidance of my advisor Professor David Kotz and did not collaborate with other researchers. I am grateful for the help from John Hudson and Arnold Song in using the Discovery cluster and acknowledge Philipp Rouast and Marc Adam for making their code and dataset publicly available to research community [57, 58].

5.1 Related work

Section 2.5 and Section 3.8 summarize related work about eating detection in the laboratory and free-living scenarios, respectively. Here we focus on existing work using CNN for action recognition in videos.

Researchers have explored various types of deep-learning architectures for action recognition in videos; four architectures are widely used, as shown in Figure 5.1. Here we give one example for each of them. Tran et al. proposed 3D CNN to address the problem of learning spatiotemporal features on large-scale video dataset [75]. They evaluated their approach on the UCF-101 dataset, which consists of 13,320 videos of 101 human action categories, and achieved 85.2% accuracy when taking red, green, blue (RGB) frames as inputs. Donahue et al. developed a CNN-long short-term memory (LSTM) model, which uses the sequence of spatial features learned by a CNN from individual video frames as input into a LSTM Recurrent Neural Network (RNN) [21]. The CNN-LSTM model has the advantage of being more flexible with regards of the number of input frames, but appears to require more training data in comparison to other approaches [57]. Simonyan et al. proposed a two-stream CNN architecture that incorporates spatial and temporal networks [68]. This architecture models short temporal snapshots of videos by averaging the predictions from a single RGB frame and a stack of 10 externally computed optical-flow frames [16]. They demonstrated that this method can achieve high performance on existing benchmarks, while being efficient to train and test [16]. Feichtenhofer et al. proposed the 'SlowFast' architecture, which involves (1) a *slow* pathway, operating at low frame rate, to capture spatial semantics, and (2) a *fast* pathway, operating at fast frame rate, to capture motion at fine temporal resolution [25]. They reported state-of-the-art accuracy on several major video-recognition benchmarks.

In recent years, researchers have explored developing ADM systems based on videos and Computer Vision (CV) techniques. Rouast et al. explored video-based intake gesture detection, which is the closest related work we were able to find in the literature [57, 58].

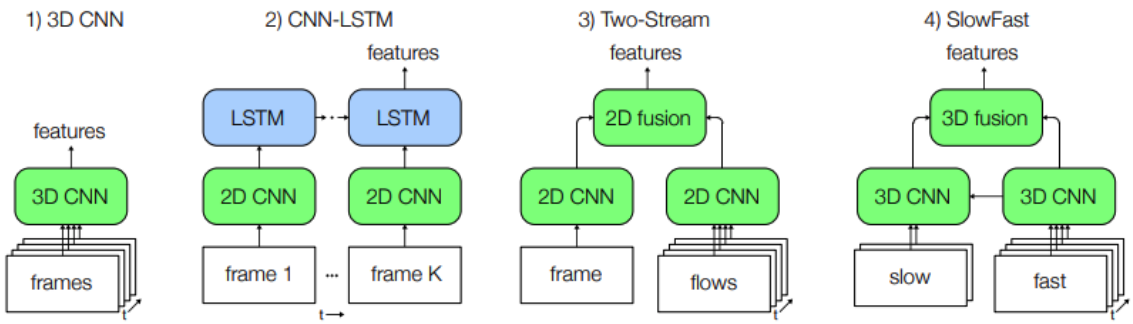

Figure 5.1: The four network architectures for action recognition in videos; figure from [57].

They placed a 360-degree camera on a dining table and recorded four participants seated around the table simultaneously while they were having a meal. In total, they collect and label video data of 102 participants in a laboratory setting. They experimented with the four different architectures in Figure 5.1 and achieved the best F1 score (0.858) with the SlowFast model.

There are three main differences between our project and their work. First, their research focused only on the detection of the intake gesture rather than the entire eating activities, which are a combination of one or more intake gestures, chewing, swallowing and other activities. In other words, the object for their work can be considered a subset of our goal. Second, we used a vision-based approach to detect eating and drinking in free-living scenarios from raw video frames of a face viewed by a wearable camera from an oblique angle, while they evaluated their method using third-person videos collected from a fixed camera in a laboratory setting. Third, their work focused on evaluating performance of intake detection with different CNN models. Other than evaluating performance of eating detection, we also validated the feasibility of deploying 3D CNN model in wearable or mobile platforms when considering computation, memory, and power constraints.

Besides, Qiu et al. proposed an approach to count the number of bites and recognize consumed food items in egocentric videos [53]. In their experiments, they achieved 74.15% top-1 accuracy (classifying between 0–4 bites in 20-second clips) for counting bites and

40.5% F1 score for recognizing 66 types of different consumed food items.

There are three main differences between our project and their work. First, our research goal is different from theirs. We focused on the detection of eating while their goal is bite counting and food recognition during known eating periods. Second, their data collection is in laboratory conditions while our study is in free-living scenarios. Third, they focused only on a performance evaluation of deep neural networks. We did both a performance evaluation and a computation, memory and power evaluation of CNN models.

5.2 System design

We refined the miniature cap-mounted camera (shown in Figure 3.6) we used for collecting ground truth in our earlier field study (Section 3.3.1) and then used it for collecting more data (videos). The resolution and frame rate of the video recorded by this camera is 360p (640×360 pixels) and 30 frames per second (FPS), which we found to be sufficient for our data analysis.

To enable longer data-collection sessions (as described in Section 5.3), we improved device comfort by adjusting the size and location of battery and the pocket holding the battery (shown in Figure 5.2), and pilot-tested various arrangements with prospective participants to find a design that fit comfortably for most, if not all, potential participants. We developed two identical sets of devices so we could collect data with one while we were sanitizing the other, or collect data in parallel on two participants.

5.3 Data collection

Under protocols approved by our IRB, we collected data using the system described in Section 5.2 in free-living scenarios.

We collected data from 10 participants (4 female, 6 male) for about 55 hours in total. During these periods of field data acquisition, participants ate various types of food includ-

Figure 5.2: Cap after adjusting the battery location

ing rice, bread, noodles, meat, vegetables, fruit, eggs, nuts, chips, soup, and ice cream. Participants recorded data in diverse environments including houses, cars, parking lots, restaurants, kitchens, woods, and streets.

After a preliminary review of data, we determined that we could not use data for about 4 hours from 2 participants, for the following reasons. For one participant, wheat flour used by the participant during cooking accidentally pasted on the camera and blurred two hours of video recorded. For another participant, the participant pressed the brim of the cap to a low position for two hours, so the camera did not capture the mouth, cheek or chin – only part of the nose. Our analysis excludes the data collected during these 4 hours. We used the remaining 51 hours of recorded video from 10 participants for further analysis. Figure 5.3 and Figure 5.4 show examples of video frames recorded during eating and non-eating periods, respectively.

The data collection protocol is similar to the study described in Section 3.3, except the participants were only asked to wear the cap; they did not wear the Auracle. Before the study, we told each participants to feel free to remove cap when they need privacy. There are two other major differences as follows.

First, due to COVID-19, Dartmouth imposed limitations on research with human subjects [20]. We thus performed many procedures of our study online (advertising, recruiting

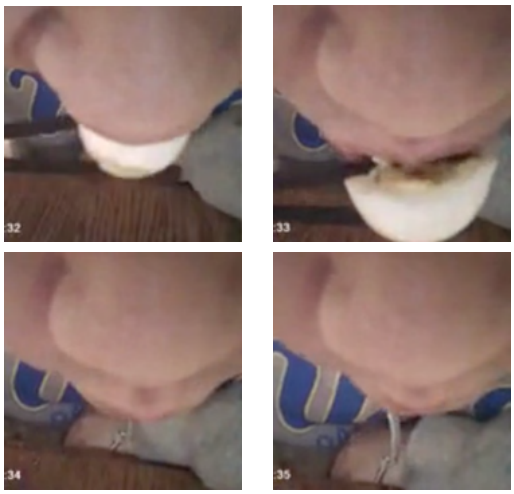

Figure 5.3: video frame examples recorded during a eating period

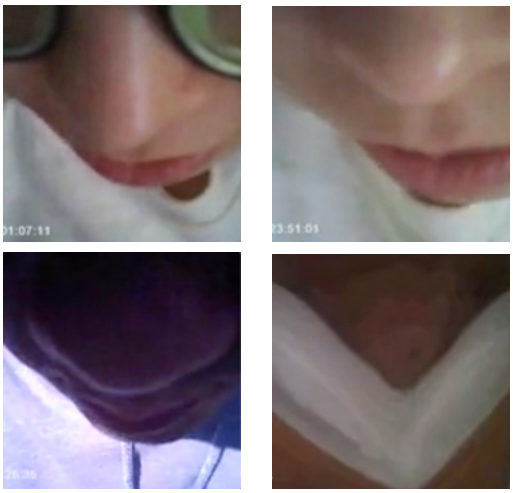

Figure 5.4: video frame examples recorded during non-eating periods

participants, meeting participants, instructing them in using the cap and performing the study, etc.). Also, we dropped off the cap before each session, and picked up the cap after each session, maintaining social distancing with participants throughout. We also sanitized the cap and camera between participants.

Second, because most people spend a relatively small fraction of their day eating, we strongly encouraged all the participants to eat more often than usual during our study. Additionally, to collect more eating data during our studies, we increased the session duration to be 5 hours (or longer), which included two meals for each participant. In our earlier study in Chapter 3, we limited the duration of each session to 2 hours because participants started to feel uncomfortable when wearing the Auracle for long periods; in this new study, most participants found the cap quite comfortable.

For video annotation, we used the same commercial service to annotate the videos as described in Section 4.3.3.

5.4 Data analysis

We next describe our evaluation metrics, and the stages of our data-processing pipeline: preprocessing, classification, and aggregation.

5.4.1 Evaluation Metrics

To better compare with the performance of the Auracle, we used the same evaluation metrics as in the window-base evaluation in Section 3.4.1. However, to reduce the computation burden of training a CNN model, we performed a global split of our video dataset rather than using a Leave-One-Person-Out (LOPO) cross-validation. We split our video dataset into 3 subsets for training, validation, and test respectively. We used the training subset for training the CNN models mentioned in Section 5.4.3, validation subset for tuning the parameters of these models, and test subset for evaluation in Section 5.5. The ratio of the total duration of videos in these three subsets is 70:15:15.

5.4.2 Data preprocessing

To reduce computational burden, we downsample the video from 30 FPS to 5 FPS, and resize from dimensions 640×360 pixels to 256×144 pixels. Because CNN models usually take inputs in square shape, and to further reduce the memory burden, we cropped the downsampled videos to extract the central 144×144 pixels.

For all the cropped videos, we used the TensorFlow library to extract all raw video frames (appearance feature) and optical flow (motion feature) and stored them in *tensorflow record* format for faster model training speed [72]. We used three RGB channels for raw video frames. We used Dual TV-L1 optical flow because it can be efficiently implemented on a modern graphics processing unit (GPU) [86]. The optical flow is calculated based on the target frame and the frame directly preceding it, and produces two channels corresponding to the horizontal and vertical components.

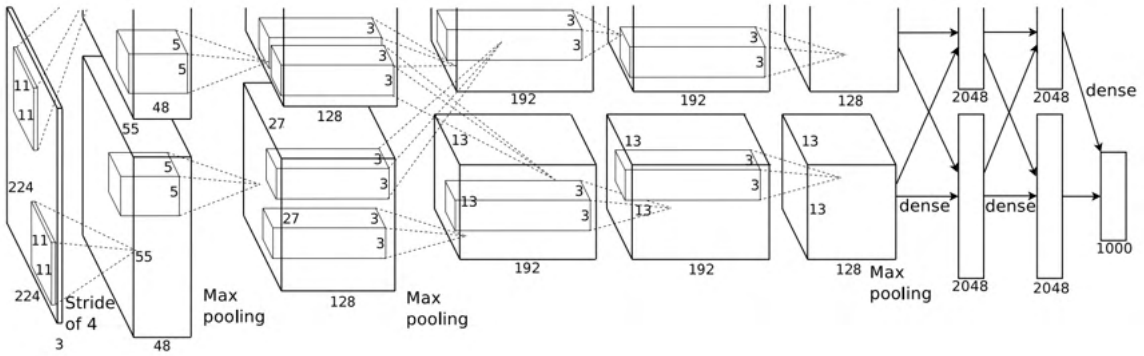

Figure 5.5: The original AlexNet model; figure from [35].

5.4.3 Classification

We developed 2-class CNN models to classify *eating* and *non-eating* using the tensorflow records we extracted in Section 5.4.2. The CNN models output a probability of *eating* for each frame (every 0.2 seconds). We run experiments with three types of CNN architectures: 2D CNN, 3D CNN, and SlowFast (see Table 5.1 for model specification). Considering the feasibility of deploying the models on wearable platform, we deliberately selected small CNN models with relatively few parameters. We adopted the five-layer CNN architecture popularised by AlexNet (Figure 5.5) for 2D CNN and 3D CNN model, which includes 4 conventional layers (each with a pooling layer after) and 1 fully connected (dense) layer [35]. For SlowFast model, there is 1 more fusion layer between the last pooling layer and the fully connected layer to combine the slow and fast pathways (see Table 5.1). We adopted and adjusted the model implementation and training policy based on the work of Rouast et al. [57].

2D CNN

We explored with two types of input features: raw video frames or precalculated optical flows. When using raw video frames as input features, the CNN model makes predictions based on the appearance information extracted from only one image segmented from videos (i.e., one video frame); the CNN model produces one inference for each frame, independently

Table 5.1: CNN model specifications. For 2D CNN, colors red and cyan show the difference between using frame and flow. For SlowFast, colors blue and *magenta* show the difference between the slow and fast pathways.

Layer	**2D CNN** (with frame or flow)			**3D CNN**			**SlowFast** (slow + *fast*)		
	dimension	kernel size	stride	dimension	kernel size	stride	dimension	kernel size	stride
data	$128^2 \times 3\|2$			$16 \times 128^2 \times 3$			$4 \times 128^2 \times 3$ $16 \times 128^2 \times 3$		
conv1	$128^2 \times 32$	3^2	1^2	$16 \times 128^2 \times 32$	3×3^2	1×1^2	$4 \times 128^2 \times 32$ $16 \times 128^2 \times 8$	$1\|3 \times 3^2$	1×1^2
pool1	$64^2 \times 32$	2^2	2^2	$8 \times 64^2 \times 32$	2×2^2	2×2^2	$4 \times 64^2 \times 32$ $16 \times 64^2 \times 8$	1×2^2	1×2^2
conv2	$64^2 \times 32$	3^2	1^2	$8 \times 64^2 \times 32$	3×3^2	1×1^2	$4 \times 64^2 \times 32$ $16 \times 64^2 \times 8$	$1\|3 \times 3^2$	1×1^2
pool2	$32^2 \times 32$	2^2	2^2	$4 \times 32^2 \times 32$	2×2^2	2×2^2	$4 \times 32^2 \times 32$ $16 \times 32^2 \times 8$	1×2^2	1×2^2
conv3	$32^2 \times 64$	3^2	1^2	$4 \times 32^2 \times 32$	3×3^2	1×1^2	$4 \times 32^2 \times 64$ $16 \times 32^2 \times 16$	$1\|3 \times 3^2$	1×1^2
pool3	$16^2 \times 64$	2^2	2^2	$1 \times 16^2 \times 64$	2×2^2	2×2^2	$4 \times 16^2 \times 64$ $16 \times 16^2 \times 16$	1×2^2	1×2^2
conv4	$16^2 \times 64$	3^2	1^2	$2 \times 16^2 \times 64$	3×3^2	1×1^2	$4 \times 16^2 \times 64$ $16 \times 16^2 \times 16$	$1\|3 \times 3^2$	1×1^2
pool4	$8^2 \times 64$	2^2	2^2	$1 \times 8^2 \times 64$	2×2^2	2×2^2	$4 \times 8^2 \times 64$ $16 \times 8^2 \times 16$	1×2^2	1×2^2
fusion							$8^2 \times 64$		
flatten	4096			4096			4096		
dense	1024			1024			1024		
dense	2			2			2		

of its classification of other frames. Because the 2D CNN model is simpler than the other two models – it uses only one frame or optical flow as the input – we anticipate it will use less memory and computation power when deploying on wearable. Additionally, 2D CNN functions as a baseline for our study, indicating what is possible with only appearance information or motion information. We used max pooling for all the pooling layers.

3D CNN

A 3D CNN has the ability to learn spatio-temporal features as it extends the 2D CNN introduced in the previous section by using 3D instead of 2D convolutions [32]. The third dimension corresponds to the temporal context. The input of 3D CNN consists the target frame and the 15 frames preceding it (3 seconds at 5 FPS), which are a sequence of 16 frames in total. In other words, the 3D CNN considers a consecutive stack of 16 video frames. The output of the CNN model is the prediction for the last frame of the sequence (the target frame). To take maximum advantage of the available training data, we generated input using a window shifting by one frame. We used temporal convolution kernels of size 3 as suggested by Tran et al. [75]. We used max pooling for temporal dimension in all the pooling layers.

SlowFast

Similar to the 3D CNN, the SlowFast model also considers a temporal context of the previous frames preceding the target frame, but the SlowFast model processes the temporal context at two different temporal resolutions. As recommended by Rouast et al. [57], we chose the factors $\alpha = 4$, temporal kernel size 3 for the fast pathway, and $\beta = 0.25$, temporal kernel size 1 for the slow pathway. We adopted the method for developing the fusion layer from the work by Feichtenhofer et al. [26].

Model training policy

We used the Adam optimizer to train each model on the training set and chose to use batch size 64 based on the memory size of the cluster we used. Training ran for 40 epochs with a learning rate starting at 2×10^{-4} and exponentially decaying at a rate of 0.9 per epoch.

We used cross entropy for loss calculation for all our models. Due to the nature of our data, the classes are imbalanced with more *non-eating* instances than *eating* instances. When training our models, we corrected this imbalance by scaling the weight of loss for each class using the reciprocal of number of instances in each class. For example, in a batch of training samples (size 64) with 54 *non-eating* instances and 10 *eating* instances, the ratio of weight of loss between *non-eating* class and *eating* class is $10 : 54$.

To avoid over fitting, we used L2 loss with a lambda of 1×10^{-4} for regularization and applied dropout in all models on convolutional and dense layers with rate 0.5. Additionally, we used early stopping if we observed the model yields increasing validation errors at the end of the training stage. We also used data augmentation by applying random transformations to the input: cropping to size 128×128, horizontal flipping, small rotations, brightness and contrast changes. Among these transformations, brightness and contrast changes can help a model better deal with eating detection in various light conditions. All models were learned end to end.

5.4.4 Aggregation

We applied the same aggregation approach as the stage-A aggregation mentioned in Section 3.4.5 and Section 3.4.6 to both prediction results and ground-truth labels. The rule we applied for aggregation in this project is the same: if more than 10% of the windows in a minute were labeled eating, we labeled that minute as eating. The difference is that the CNN models output predictions every 0.2 seconds (one prediction per frame) while the classifiers we used in Section 3.4.4 output predictions every 3 seconds (one prediction per 3-second time window). After aggregation, the resolution of eating-detection results are 1 minute in

both cases.

5.5 Performance evaluation

Table 5.2 summarizes the resulting performance metrics for eating detection with a 1-minute resolution using the four models. We achieved the best result using SlowFast model, with an F1 score of 78.7% and accuracy of 90.9%.

To assess the usefulness of temporal context, we compare the accuracy of our models with and without temporal context. Based on Table 5.2, the 3D CNN model (F1 score 73.8%) outperforms 2D CNN (with frame; F1 score 43.3%) and 2D CNN (with flow; F1 score 55.4%). The SlowFast model also outperforms 2D CNN (with frame) and 2D CNN (with flow) by more than 23% F1 score. We thus conclude that (1) temporal context is crucial for eating detection in the field and considerably improves model performance; (2) using only spatial information (either frame (appearance) or flow (motion) feature) from one single video frame may be not sufficient for achieving good eating-detection performance.

Additionally, we noticed that precision is the worst score across all the metrics for all the four models we experimented. The low precision score indicates that there are many false positives (the model indicated eating and ground truth indicated non-eating) in the predictions of our models. To identify the reasons, we checked the video frames during the periods that false positives occurred. Some scenarios where false positives occurred include talking, drinking, blowing nose, putting on face masks, mouth rinsing, wiping mouth with napkin, unconscious mouth or tongue movement, and continuously touching face or mouth. We anticipate more training data and deeper CNN networks would help to reduce false positives.

Table 5.2: Performance metrics for eating detection with CNN models.

Model	#Parameters	Accuracy	Precision	Recall	F1 Score
2D CNN (with frame)	4.26M	71.0%	38.3%	49.8%	43.3%
2D CNN (with flow)	4.26M	78.3%	46.9%	67.8%	55.4%
3D CNN	4.39M	86.4%	72.4%	75.3%	73.8%
SlowFast	4.49M	90.9%	75.5%	82.2%	78.7%

5.6 Computation, memory, and power evaluation

Based on the performance evaluation in Section 5.5, we found both the 3D CNN and SlowFast models achieved better performance than the 2D CNN models for eating detection. However, the SlowFast model is a fusion of two 3D CNN models so we assume it requires more computational resources than a single 3D CNN model. In this section, we thus focus on whether it is feasible to deploy the 3D CNN model on a mobile or wearable platform, when considering computation, memory, and power constraints.

The computational resources needed for a deep-learning model is often measured in *gigaflops:* 1×10^9 floating point operations per second (GFLOPs). Niu et al. measured a 3D CNN model having 8 convolutional layers and found the overall model requires from 10.8 to 15.2 GFLOPs, after compression with different pruning algorithms [45]. We used a 3D CNN model with 4 convolutional layers and we assume our model would thus require less than 10.8 GFLOPs after pruning.

We then investigated GPUs used in modern mobile or wearable platforms. The Google Pixel 3 smartphone has a Qualcomm Adreno 630 GPU that can support 727 GFLOPs [82]. Many modern smartwatches and similar wearable platforms have GPUs as well. For instance, the Huawei Watch GT 2 includes a Qualcomm Adreno 304 GPU that supports 19.2 GFLOPs [82]. Both these platforms have enough computing resources to run our 3D CNN model for inference; we thus conclude that modern mobile or wearable platforms can support the models described in Section 5.4.3.

The memory needed for running the 3D CNN models include at least two parts: storing the raw video frame sequence, and storing the model parameters. The pixel values of RGB

images are integers and the model parameters are floating-point numbers, which (in our implementation) are 4 bytes each. Using the data dimensions from Table 5.1, the memory needed for storing the raw video frame sequence is $16 \times 128^2 \times 3 \times 4 = 3.15$ MB. Using the parameters from Table 5.2, the memory needed for storing the parameters of the 3D CNN model is $4.39 \times 4 = 17.56$ MB. Hence the memory needed for running the 3D CNN model is at about $3.15 + 17.56 = 20.71$ MB, and should fit easily in a mobile platform with 32 MB of main memory. Such platforms are readily available and suitable for small wearable devices today. (For instance, the Apple Watch series 6 has 1000 MB RAM [83].)

The power consumption of the system consists of at least two parts: the camera (to capture images or videos) and the processor (to run the CNN model). We investigated ultra-low power CMOS cameras in the literature; a camera with parameters similar to ours (96×96 pixels, 20 FPS) consumes less than $20\,\mu\mathrm{W}$ [17]. We conclude that the power consumption for capturing images or videos can be ignored, if using an ultra-low power CMOS camera that is specifically designed as needed.

We found little information, however, regarding the power consumption of GPUs used in mobile or wearable platforms (e.g., Qualcomm Adreno 304) in literature or online. We were only able to find that mobile GPUs are typically designed for a power ceiling under $1\,\mathrm{W}$ [18]. Given this assumption, the upper limit of power consumption for continuous running the 3D CNN model for a waking day (16 hours) is $16\,\mathrm{W\,h}$, which is $4234\,\mathrm{mA\,h}$ when the voltage is $3.7\,\mathrm{V}$. To address this need, we could use two *18650 lithium-ion batteries* (e.g., Samsung 35E 18650 battery) as the power supply for the system, which have enough capacity ($7000\,\mathrm{mA\,h}$ in total) and are cheap ($5–10 each), small (18.75 mm $\times$ 65.25 mm), and rechargeable [3]. Note that this is estimate is only a rough upper bound on GPU power consumption. A GPU that can run our model does not need to be powered at $1\,\mathrm{W}$ and the GPU does not necessarily need to continuously run for 16 hours. For instance, during some periods users may be sitting quietly at a desk while studying, so there is no movement captured by the camera. These periods can be easily filtered out as *non-eating*

and we can set the GPU to idle mode to save power, much like the lower-power 'sleep' state discussed in Section 3.6.

Because modern mobile phones often have a powerful GPU, it maybe beneficial to transmit the video frames from the cap to the mobile phone for running 3D CNN models – assuming current Bluetooth technology can support the necessary data-transmission rate. The video we used is 5 FPS so our system needs to transmit $5 \times 128^2 \times 3 \times 4 = 0.98$ MB per second, which is about 8 megabit per second (Mbps). The new Bluetooth 5.0 technology can support a data transfer rate as high as 50 Mbps, so it may indeed feasible to take this approach [15]. Further investigation would be necessary to consider the power tradeoff between on-board GPU processing vs. Bluetooth transfer to the phone for processing. Privacy is another potential issue, as the export of raw video from the cap to the phone poses a potential risk for that video being obtained by network eavesdroppers or malware based in the phone.

5.7 Future work

Here we discuss ideas that we think deserve exploring in the future.

Detection of drinking and other health-related behaviours. CNN models have been widely used for the recognition of various human actions in videos [16,75,76]. With enough training data and proper model tuning, our method has great potential to generalize to the detection of other health-related behaviours (such as drinking, smoking, coughing, sniffling, laughing, breathing, speaking, and face touching). However, we note that most of these behaviors are usually short and infrequent during normal daily life, so large-scale field studies (and substantial video annotation effort) may be necessary to collect enough training data.

Images and videos with different key parameters. In this project, we only experimented with RGB videos frames that have relatively low resolution (144×144 pixels) and low frame rate (5 FPS) due to limited computation resource. In the future, it would be interesting to explore different key parameters (i.e., frame rate, frame resolution, color depth) that affect cost (e.g., power consumption) and performance (e.g., F1 score) of the approaches we used, and characterize the trade-offs between cost and performance as these parameters change.

Fusion of visual and privacy-sensitive audio signals. Researchers have developed many acoustic-based ADM systems for eating detection and showed that audio signals (e.g., chewing sound and swallowing sound) are useful for eating detection [5, 12, 88]. Our system is located close to the face and can be easily modified to capture both video and audio signals. In our experiment, we chose not to collect audio, due to privacy concerns. An on-board module that could process audio on the fly could be useful to address this issue [79]. Thus, it is worth investigating the fusion of visual and privacy-sensitive audio signals, which may yield better performance in eating detection.

Deeper CNN networks. If experimenting with deeper CNN networks, the performance of eating detection may further improve. There exist implementations of many pre-trained deeper networks, such as ResNet and GoogleNet, that could be used to initialize a model that could then be fine-tuned for eating detection [29, 70]. Specifically, it is worth exploring these deeper networks as a backbone for the 3D CNN and SlowFast models, to see how much improvement the deeper networks can achieve.

Different types of cameras. In this study, we developed an eating-detection approach using a traditional digital camera and CV techniques. Other types of cameras (e.g., thermal cameras and event cameras) could also be useful sensors for eating detection. Thermal cameras could take advantage of the temperature information from food and used it as an useful cue for eating detection. Event cameras contain pixels that independently respond to

changes in brightness as they occur [38]. Compared with traditional cameras, event cameras have several benefits including extremely low latency, asynchronous data acquisition, high dynamic range, and very low power consumption [92], which make them interesting sensors to explore for eating detection in the future.

Explainability of CNN model. The development of methods for visualizing, explaining and interpreting deep learning models has recently attracted increasing attention [60]. One of the most popular methods is the use of heatmaps to visualize the importance of each pixel for the prediction. Similar explanation methods could help us to understand the reason our models arrived at a specific decision, so we could further improve our eating-detection approaches accordingly.

5.8 Summary

In this chapter, we developed a computer-vision based approach for eating detection. Indeed, we believe this chapter represents the first work to experiment with video-based eating detection in free-living scenarios. Using a miniature head-mounted camera, we conducted a field study and collected data with 10 participants for about 55 hours. We designed a data-processing pipeline and evaluated performance of eating detection using four different CNN models. The best model achieved an accuracy 90.9% and an F1 score 78.7% for eating detection with a 1-minute resolution. Finally, we discussed the feasibility of deploying the 3D CNN model in wearable or mobile platforms when considering computation, memory, and power constraints.

6

Summary

In this dissertation, we present our work in detecting health-related behaviors using head-mounted devices. Using an acoustic approach, we demonstrated methods for eating detection in laboratory conditions and then explored the generalization of our work along two dimensions: from laboratory conditions to free-living conditions, and from adult population to children population. First, based on the method we experimented in laboratory conditions, we developed *Auracle*, a wearable earpiece that can automatically detect eating in free-living conditions. We collected data with 14 participants for 32 hours in a field study and achieved accuracy exceeding 92.8% and F1 score exceeding 77.5% for eating detection. Second, in children's eating studies, we adapted *Auracle* to allow data collection from children and

improved the accuracy and robustness of the eating-activity detection algorithms. Using the improved prototype, we conducted a lab study with a sample of 10 children for 60 total sessions and collected 22.3 hours of data in both meal and snack scenarios. We achieved a 95.5% accuracy and a 95.7% F1 score for eating detection in laboratory conditions.

In addition, we developed a computer-vision based approach for eating detection in free-living scenarios. Using a miniature head-mounted camera, we conducted a field study and collected data with 10 participants for about 55 hours. We explored using deep-learning models (CNN) rather than the statistical machine-learning models we used in acoustic approaches. The best model achieved an accuracy 90.9% and a F1 score 78.7% for eating detection. The overall work aims to detect *when* and *how long* people perform health-related behaviors, which are the foundation for ADM and monitoring of other health-related behaviors (e.g., alcohol consumption monitoring), and thus support and benefit health-science research.

Bibliography

[1] O. Amft, M. Stäger, P. Lukowicz, and G. Tröster. Analysis of Chewing Sounds for Dietary Monitoring. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2005. DOI 10.1007/11551201_4. Citation on pages 10, 21, 24, 26, and 58.

[2] O. Amft and G. Troster. On-Body Sensing Solutions for Automatic Dietary Monitoring. *IEEE Pervasive Computing*, 8(2) pages 62–70, 4 2009. DOI 10.1109/mprv.2009.32. Citation on page 53.

[3] 18650 battery store. Online at https://www.18650batterystore.com/collections/samsung-18650-batteries, visited April 2021. Citation on page 91.

[4] A. Bedri, D. Li, R. Khurana, K. Bhuwalka, and M. Goel. FitByte: Automatic Diet Monitoring in Unconstrained Situations Using Multimodal Sensing on Eyeglasses CCS Concepts. In *CHI Conference on Human Factors in Computing Systems*, volume 20, pages 1–12, 2020. DOI 10.1145/3313831.3376869. Citation on page 78.

[5] 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. *Proc. ACM Interactive, Mobile and Wearable Ubiquitous*

Technology, 1(3), 9 2017. DOI 10.1145/3130902. Citation on pages 24, 32, 54, 78, and 93.

[6] A. Bedri, A. Verlekar, E. Thomaz, V. Avva, and T. Starner. A wearable system for detecting eating activities with proximity sensors in the outer ear. In *Proceedings of the ACM International Symposium on Wearable Computers*, pages 91–92. ACM, 2015. DOI doi:10.1145/2802083.2808411. Citation on page 54.

[7] A. Bedri, A. Verlekar, E. Thomaz, V. Avva, and T. Starner. Detecting Mastication: A Wearable Approach. In *Proceedings of the ACM on International Conference on Multimodal Interaction*, 2015. DOI 10.1145/2818346.2820767. Citation on pages 10 and 24.

[8] A. Bedri, A. Verlekar, E. Thomaz, V. Avva, and T. Starner. Detecting Mastication: A Wearable Approach. In *Proceedings of the ACM on International Conference on Multimodal Interaction*, 2015. DOI 10.1145/2818346.2820767. Citation on page 54.

[9] B. M. Bell, R. Alam, N. Alshurafa, J. Lach, and D. Spruijt-metz. Automatic , wearable-based , in-field eating detection approaches for public health research : a scoping review. *npj Digital Medicine*, 2020. DOI 10.1038/s41746-020-0246-2. Citation on page 78.

[10] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29(4) pages 1165–1188, 2001. DOI 10.1214/aos/1013699998. Citation on pages 15 and 41.

[11] S. Bi, T. Wang, E. Davenport, R. Peterson, R. Halter, J. Sorber, and D. Kotz. Toward a Wearable Sensor for Eating Detection. In *Proceedings of the ACM Workshop on Wearable Systems and Applications (WearSys)*, pages 17–22. ACM Press, 6 2017. DOI 10.1145/3089351.3089355. Citation on pages 24, 29, 57, and 58.

[12] S. Bi, T. Wang, N. Tobias, J. Nordrum, S. Wang, G. Halvorsen, S. Sen, R. Peterson, K. Odame, K. Caine, R. Halter, J. Sorber, and D. Kotz. Auracle: Detecting Eating Episodes with an Ear-Mounted Sensor. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) (Ubicomp)*, 2(3), 9 2018. DOI 10.1145/3264902. Citation on pages 58 and 93.

[13] T. Bodenheimer, E. Chen, and H. D. Bennett. Confronting the growing burden of chronic disease: Can the U.S. health care workforce do the job?, 1 2009. DOI 10.1377/hlthaff.28.1.64. Citation on page 1.

[14] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. Essentia: An audio analysis library for music information retrieval. *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013*, pages 493–498, 2013. DOI 10.1145/2502081.2502229. Citation on page 68.

[15] Bluetooth: everything you need to know about the popular wireless standard. Online at https://www.ionos.com/digitalguide/server/know-how/bluetooth/, visited April 2021. Citation on page 92.

[16] J. Carreira and A. Zisserman. Quo Vadis, action recognition? A new model and the kinetics dataset. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, v.2017-Janua pages 4724–4733, 2017. DOI 10.1109/CVPR.2017.502. Citation on pages 77, 79, and 92.

[17] I. Cevik, X. Huang, H. Yu, M. Yan, and S. Ay. An Ultra-Low Power CMOS Image Sensor with On-Chip Energy Harvesting and Power Management Capability. *Sensors*, 15(3) pages 5531–5554, 3 2015. DOI 10.3390/s150305531. Citation on page 91.

[18] K. T. Cheng and Y. C. Wang. Using mobile GPU for general-purpose computing a case study of face recognition on smartphones. In *Proceedings of 2011 International*

Symposium on VLSI Design, Automation and Test, VLSI-DAT 2011, pages 54–57, 2011. DOI 10.1109/VDAT.2011.5783575. Citation on page 91.

[19] K. S. Chun, S. Bhattacharya, and E. Thomaz. Detecting Eating Episodes by Tracking Jawbone Movements with a Non-Contact Wearable Sensor. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1) pages 1–21, 3 2018. DOI 10.1145/3191736. Citation on pages 23, 54, and 58.

[20] D. College. Policy on conduct of human subjects research activities during covid-19 operations. Online at https://www.dartmouth-hitchcock.org/sites/default/files/2021-02/policy-on-conduct-of-human-research-activities-during-covid.pdf, visited April 2021. Citation on page 82.

[21] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4) pages 677–691, 2017. DOI 10.1109/TPAMI.2016.2599174. Citation on page 79.

[22] 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*, 18(4) pages 1253–1260, 7 2014. DOI 10.1109/JBHI.2013.2282471. Citation on pages 26, 39, and 59.

[23] M. Farooq, J. M. Fontana, and E. Sazonov. A novel approach for food intake detection using electroglottography. *Physiological Measurement*, 35(5) pages 739–751, 2014. DOI 10.1088/0967-3334/35/5/739. Citation on page 22.

[24] M. Farooq and E. Sazonov. A Novel Wearable Device for Food Intake and Physical Activity Recognition. *Sensors*, 16(7), 7 2016. DOI 10.3390/s16071067. Citation on page 22.

[25] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. *Proceedings of the IEEE International Conference on Computer Vision*, v.2019-Octob pages 6201–6210, 2019. DOI 10.1109/ICCV.2019.00630. Citation on page 79.

[26] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional Two-Stream Network Fusion for Video Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. DOI 10.1109/CVPR.2016.213. Citation on page 87.

[27] R. Friedrich, S. Siegert, J. Peinke, S. Lück, M. Siefert, M. Lindemann, J. Raethjen, G. Deuschl, and G. Pfister. Extracting model equations from experimental data. *Physics Letters, Section A: General, Atomic and Solid State Physics*, 271(3) pages 217–222, 6 2000. DOI 10.1016/S0375-9601(00)00334-0. Citation on page 20.

[28] C. M. Hales, M. D. Carroll, C. D. Fryar, and C. L. Ogden. Prevalence of Obesity Among Adults and Youth: United States, 2015–2016. NCHS data brief, no 288. Hyattsville, MD: National Center for Health Statistics. *NCHS data brief, no 288. Hyattsville, MD: National Center for Health Statistics.*, 2017. Online at https://www.cdc.gov/nchs/products/databriefs/db288.htm. Citation on page 56.

[29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, pages 770–778. IEEE Computer Society, 12 2016. DOI 10.1109/CVPR.2016.90. Citation on page 93.

[30] H. Kalantarian, N. Alshurafa, and M. Sarrafzadeh. A wearable nutrition monitoring system. In *Proceedings - 11th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2014*, pages 75–80. IEEE Computer Society, 2014. DOI 10.1109/BSN.2014.26. Citation on pages 10, 22, and 24.

[31] H. Kalantarian, N. Alshurafa, and M. Sarrafzadeh. A Survey of Diet Monitoring Technology. *IEEE Pervasive Computing*, 16(1) pages 57–65, 1 2017. DOI 10.1109/mprv.2017.1. Citation on page 53.

[32] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale Video Classification with Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 1725–1732, 2014. DOI 10.1109/CVPR.2014.223. Citation on pages 77 and 87.

[33] R. C. Klesges, T. J. Coates, G. Brown, J. Sturgeon-Tillisch, L. M. Moldenhauer-Klesges, B. Holzer, J. Woolfrey, and J. Vollmer. Parental influences on children's eating behavior and relative weight. *Journal of applied behavior analysis*, 16(4) pages 371–378, 1983. DOI 10.1901/jaba.1983.16-371. Citation on page 75.

[34] T. V. Kral and E. M. Rauh. Eating behaviors of children in the context of their family environment. *Physiology and Behavior*, 100(5) pages 567–573, 2010. DOI 10.1016/j.physbeh.2010.04.031. Citation on page 57.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6) pages 84–90, 6 2017. DOI 10.1145/3065386. Citation on page 85.

[36] F. Lecluse, M. Brocaar, and J. Verschuure. The Electroglottography and its Relation to Glottal Activity. *Folia Phoniatrica et Logopaedica*, 27(3) pages 215–224, 1975. DOI 10.1159/000263988. Citation on page 22.

[37] R. M. Leech, A. Worsley, A. Timperio, and S. A. McNaughton. Characterizing eating patterns: a comparison of eating occasion definitions. *The American Journal of Clinical Nutrition*, 10 2015. DOI 10.3945/ajcn.115.114660. Citation on page 26.

[38] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128 × 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid State Circuits*, 43(2) pages 566–576, 2008. DOI 10.1109/JSSC.2007.914337. Citation on page 94.

[39] J. Liu, E. Johns, L. Atallah, C. Pettitt, B. Lo, G. Frost, and G.-Z. Yang. An Intelligent Food-Intake Monitoring System Using Wearable Sensors. In *Ninth International Conference on Wearable and Implantable Body Sensor Networks*, pages 154–160. Hamlyn Centre, Imperial Coll. London, London, UK, 5 2012. DOI 10.1109/bsn.2012.11. Citation on pages 21, 26, and 58.

[40] C. H. Llewellyn, C. H. Van Jaarsveld, D. Boniface, S. Carnell, and J. Wardle. Eating rate is a heritable phenotype related to weight in children. *American Journal of Clinical Nutrition*, 88(6) pages 1560–1566, 2008. DOI 10.3945/ajcn.2008.26175. Citation on page 76.

[41] C. Merck, C. Maher, M. Mirtchouk, M. Zheng, Y. Huang, and S. Kleinberg. Multimodality Sensing for Eating Recognition. In *Proceedings of the EAI International Conference on Pervasive Computing Technologies for Healthcare*. ACM Press, 2016. DOI 10.4108/eai.16-5-2016.2263281. Citation on page 23.

[42] 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 (IMWUT)*, 1(3) pages 85+, 9 2017. DOI 10.1145/3131894. Citation on page 54.

[43] M. Mirtchouk, C. Merck, and S. Kleinberg. Automated estimation of food type and amount consumed from body-worn audio and motion sensors. In *UbiComp 2016 - Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 451–462, New York, NY, USA, 9 2016. Association for Computing Machinery, Inc. DOI 10.1145/2971648.2971677. Citation on page 24.

[44] J. Müller, A. M. Hoch, V. Zoller, and R. Oberhoffer. Feasibility of physical activity assessment with wearable devices in children aged 4-10 Years-A Pilot study. *Frontiers in Pediatrics*, 6(January) pages 1–5, 2018. DOI 10.3389/fped.2018.00005. Citation on page 57.

[45] W. Niu, M. Sun, Z. Li, J.-A. Chen, J. Guan, X. Shen, Y. Wang, S. Liu, X. Lin, and B. Ren. RT3D: Achieving Real-Time Execution of 3D Convolutional Neural Networks on Mobile Devices. *arXiv*, 7 2020. Online at http://arxiv.org/abs/2007.09835. Citation on page 90.

[46] T. Olubanjo and M. Ghovanloo. Real-time swallowing detection based on tracheal acoustics. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4384–4388, 5 2014. DOI 10.1109/icassp.2014.6854430. Citation on pages 22, 26, and 58.

[47] V. Papapanagiotou, C. Diou, and A. Delopoulos. Chewing detection from an in-ear microphone using convolutional neural networks. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 1258–1261, 2017. DOI 10.1109/EMBC.2017.8037060. Citation on page 58.

[48] V. Papapanagiotou, C. Diou, L. Zhou, J. van den Boer, M. Mars, and A. Delopoulos. A novel chewing detection system based on PPG, audio and accelerometry. *IEEE Journal of Biomedical and Health Informatics*, 2016. DOI 10.1109/jbhi.2016.2625271. Citation on pages 10, 23, 24, 39, and 66.

[49] 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*, 33(6) pages 1073–1093, 6 2012. DOI 10.1088/0967-3334/33/6/1073. Citation on pages 21, 26, and 58.

[50] M. Patel, D. Asch, and K. Volpp. Wearable devices as facilitators, not drivers, of health behavior change. BT - J Am Med Assoc. 2015;313:459?460 (Published online. *JAMA : the journal of the American Medical Association*, 19104(5) pages 2–3, 2015. DOI 10.1001/jama.2014.14781.Conflict. Citation on page 2.

[51] M. H. Pesch and J. C. Lumeng. Methodological considerations for observational coding of eating and feeding behaviors in children and their families. *International Journal of Behavioral Nutrition and Physical Activity*, 14(1) pages 1–14, 2017. DOI 10.1186/s12966-017-0619-3. Citation on page 57.

[52] T. Prioleau, E. Moore, and M. Ghovanloo. Unobtrusive and Wearable Systems for Automatic Dietary Monitoring. *IEEE Transactions on Biomedical Engineering*, 64(9) pages 2075–2089, 9 2017. DOI 10.1109/tbme.2016.2631246. Citation on page 53.

[53] J. Qiu, F. P. W. Lo, S. Jiang, C. Tsai, Y. Sun, and B. Lo. Counting Bites and Recognizing Consumed Food from Videos for Passive Dietary Monitoring. *IEEE Journal of Biomedical and Health Informatics*, pages 1–1, 9 2020. DOI 10.1109/jbhi.2020.3022815. Citation on page 80.

[54] W. Raghupathi and V. Raghupathi. An empirical study of chronic diseases in the united states: A visual analytics approach. *International Journal of Environmental Research and Public Health*, 15(3), 3 2018. DOI 10.3390/ijerph15030431. Citation on page 1.

[55] T. Rahman, A. T. Adams, M. Zhang, E. Cherry, B. Zhou, H. Peng, and T. Choudhury. BodyBeat: A Mobile System for Sensing Non-speech Body Sounds. In *Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2014. DOI 10.1145/2594368.2594386. Citation on pages 10, 11, 15, 22, 26, and 58.

[56] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen. Image browsing, processing, and clustering for participatory sensing. In *Proceedings of the 4th workshop*

on Embedded networked sensors - EmNets '07, pages 13–17, 2007. DOI 10.1145/1278972.1278975. Citation on pages 26 and 59.

[57] P. V. Rouast and M. Adam. Learning deep representations for video-based intake gesture detection. *IEEE Journal of Biomedical and Health Informatics*, PP(8) pages 1–1, 2019. DOI 10.1109/jbhi.2019.2942845. Citation on pages 78, 79, 80, 85, and 87.

[58] P. V. Rouast, H. Heydarian, M. T. Adam, and M. E. Rollo. OReBA: A dataset for objectively recognizing eating behavior and associated intake. *IEEE Access*, v.8 pages 181955–181963, 2020. DOI 10.1109/ACCESS.2020.3026965. Citation on pages 78 and 79.

[59] D. Ruta and B. Gabrys. An Overview of Classifier Fusion Methods. *Computing and Information systems*, 7(1) pages 1–10, 2000. Online at http://dec.bournemouth.ac.uk/staff/bgabrys/publications/CIS_2000_Ruta_Gabrys_fusion_methods_overview.pdf. Citation on page 52.

[60] W. Samek, T. Wiegand, and K.-R. Müller. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. *ITU Journal: ICT Discoveries*, 1(No.1) pages 39–48, 2018. Online at https://www.itu.int/en/journal/001/Pages/05.aspx. Citation on page 94.

[61] 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*, 29(5) pages 525–541, 5 2008. DOI 10.1088/0967-3334/29/5/001. Citation on pages 21, 22, 26, and 58.

[62] 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 Bio-medical*

Engineering, 57(3) pages 626–633, 3 2010. DOI 10.1109/TBME.2009.2033037. Citation on pages 21, 26, and 58.

[63] J. B. Schwimmer, T. M. Burwinkle, and J. W. Varni. Health-Related Quality of Life of Severely Obese Children and Adolescents. *Journal of the American Medical Association*, 289(14) pages 1813–1819, 2003. DOI 10.1001/jama.289.14.1813. Citation on page 56.

[64] S. Sen, V. Subbaraju, A. Misra, R. K. Balan, and Y. Lee. The case for smartwatch-based diet monitoring. In *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 585–590, 3 2015. DOI 10.1109/percomw.2015.7134103. Citation on pages 26, 54, and 59.

[65] S. Sen, V. Subbaraju, A. Misra, R. K. Balan, and Y. Lee. Experiences in Building a Real-World Eating Recogniser. In *Proceedings of the 4th International on Workshop on Physical Analytics*, WPA, pages 7–12, 2017. DOI 10.1145/3092305.3092306. Citation on page 54.

[66] 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*, 21(3) pages 599–606, 5 2017. DOI 10.1109/JBHI.2016.2612580. Citation on page 10.

[67] M. Shuzo, S. Komori, T. Takashima, G. Lopez, S. Tatsuta, S. Yanagimoto, S. Warisawa, J.-J. Delaunay, and I. Yamada. Wearable Eating Habit Sensing System Using Internal Body Sound. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 4(1) pages 158–166, 2010. DOI 10.1299/jamdsm.4.158. Citation on pages 22, 26, and 58.

[68] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in neural information processing systems*, pages 568–576, 2014. DOI 10.5555/2968826.2968890. Citation on pages 77 and 79.

[69] M. Sun, L. E. Burke, Z. H. Mao, Y. Chen, H. C. Chen, Y. Bai, Y. Li, C. Li, and W. Jia. eButton: A Wearable Computer for Health Monitoring and Personal Assistance. In *Proceedings of the Annual Design Automation Conference*, 2014. DOI 10.1145/2593069.2596678. Citation on pages 26 and 59.

[70] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June-2015, pages 1–9. IEEE Computer Society, 10 2015. DOI 10.1109/CVPR.2015.7298594. Citation on page 93.

[71] A. Tada and H. Miura. Association of mastication and factors affecting masticatory function with obesity in adults: A systematic review. *BMC Oral Health*, 18(1) pages 1–8, 2018. DOI 10.1186/s12903-018-0525-3. Citation on page 58.

[72] TensorFlow. Tensorflow website. Online at https://www.tensorflow.org/, visited April 2021. Citation on page 84.

[73] E. Thomaz, A. Parnami, I. Essa, and G. D. Abowd. Feasibility of identifying eating moments from first-person images leveraging human computation. In *Proceedings of the 4th International SenseCam & Pervasive Imaging Conference on - SenseCam*, pages 26–33. ACM, ACM Press, 2013. DOI 10.1145/2526667.2526672. Citation on pages 26 and 59.

[74] E. Thomaz, C. Zhang, I. Essa, and G. D. Abowd. Inferring meal eating activities in real world settings from ambient sounds: A feasibility study. *International Conference*

on Intelligent User Interfaces, Proceedings IUI, v.2015-Janua pages 427–431, 2015. DOI 10.1145/2678025.2701405. Citation on page 54.

[75] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*, v.2015 Inter pages 4489–4497, 2015. DOI 10.1109/ICCV.2015.510. Citation on pages 77, 79, 87, and 92.

[76] D. Tran, H. Wang, L. Torresani, J. Ray, Y. Lecun, and M. Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. DOI 10.1109/CVPR.2018.00675. Citation on pages 77 and 92.

[77] J. van den Boer, A. van der Lee, L. Zhou, V. Papapanagiotou, C. Diou, A. Delopoulos, and M. Mars. The SPLENDID Eating Detection Sensor: Development and Feasibility Study. *JMIR mHealth and uHealth*, 6(9) page e170, 2018. DOI 10.2196/mhealth.9781. Citation on page 58.

[78] T. Vu, F. Lin, N. Alshurafa, and W. Xu. Wearable Food Intake Monitoring Technologies: A Comprehensive Review. *Computers*, 6(1), 2017. DOI 10.3390/computers6010004. Citation on page 53.

[79] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell. Studentlife: Assessing mental health, academic performance and behavioral trends of college students using smartphones. In *UbiComp 2014 - Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 3–14, New York, NY, USA, 9 2014. Association for Computing Machinery, Inc. DOI 10.1145/2632048.2632054. Citation on page 93.

[80] J. A. Ward, P. Lukowicz, and H. W. Gellersen. Performance metrics for activity recognition. *ACM Transactions on Intelligent Systems and Technology*, 2(1) pages 1–23, 1 2011. DOI 10.1145/1889681.1889687. Citation on pages 39, 40, and 48.

[81] Z. J. Ward, M. W. Long, S. C. Resch, C. M. Giles, A. L. Cradock, and S. L. Gortmaker. Simulation of growth trajectories of childhood obesity into adulthood. *New England Journal of Medicine*, 377(22) pages 2145–2153, 2017. DOI 10.1056/NEJMoa1703860. Citation on page 56.

[82] Wikipedia. Wikipedia for adreno GPU. Online at https://en.wikipedia.org/wiki/Adreno, visited April 2021. Citation on page 90.

[83] Wikipedia. Wikipedia for Apple Watch series 6. Online at https://en.wikipedia.org/wiki/Apple_Watch_Series_6, visited April 2021. Citation on page 91.

[84] C. C. Yang and Y. L. Hsu. A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors*, 10(8) pages 7772–7788, 2010. DOI 10.3390/s100807772. Citation on page 57.

[85] K. Yatani and K. N. Truong. BodyScope: a wearable acoustic sensor for activity recognition. In *Proceedings of the ACM Conference on Ubiquitous Computing (UbiComp)*, Ubicomp, pages 341–350, 9 2012. DOI 10.1145/2370216.2370269. Citation on pages 22, 26, and 58.

[86] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *DAGM conference on Pattern recognition*, pages 214–223, 2007. Online at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.709.4597&rep=rep1&type=pdf. Citation on page 84.

[87] R. Zhang and O. Amft. Bite Glasses: Measuring Chewing Using EMG and Bone Vibration in Smart Eyeglasses. In *Proceedings of the ACM International Symposium on Wearable Computers*, 2016. DOI 10.1145/2971763.2971799. Citation on page 22.

[88] R. Zhang and O. Amft. Monitoring Chewing and Eating in Free-Living Using Smart Eyeglasses. *IEEE Journal of Biomedical and Health Informatics*, 22(1) pages 23–32, 1 2018. DOI 10.1109/jbhi.2017.2698523. Citation on pages 24, 54, and 93.

[89] R. Zhang and O. Amft. Retrieval and Timing Performance of Chewing-Based Eating Event Detection in Wearable Sensors. *Sensors*, 20(2) page 557, 1 2020. DOI 10.3390/s20020557. Citation on page 58.

[90] R. Zhang, S. Bernhart, and O. Amft. Diet eyeglasses: Recognising food chewing using EMG and smart eyeglasses. In *BSN 2016 - 13th Annual Body Sensor Networks Conference*, pages 7–12. Institute of Electrical and Electronics Engineers Inc., 7 2016. DOI 10.1109/BSN.2016.7516224. Citation on pages 22 and 24.

[91] S. Zhang, Y. Zhao, D. T. Nguyen, R. Xu, S. Sen, J. Hester, and N. Alshurafa. NeckSense: A Multi-Sensor Necklace for Detecting Eating Activities in Free-Living Conditions. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) (Ubicomp)*, 4(2), 6 2020. DOI 10.1145/3397313. Citation on pages 24 and 78.

[92] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis. The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. *IEEE Robotics and Automation Letters*, 3(3) pages 2032–2039, 7 2018. DOI 10.1109/LRA.2018.2800793. Citation on page 94.