Detecting Meals In the Wild Using the Inertial Data of a Typical Smartwatch

Konstantinos Kyritsis, Christos Diou and Anastasios Delopoulos

Abstract— **Automated and objective monitoring of eating behavior has received the attention of both the research community and the industry over the past few years. In this paper we present a method for automatically detecting meals in free living conditions, using the inertial data (acceleration and orientation velocity) from commercially available smartwatches. The proposed method operates in two steps. In the first step we process the raw inertial signals using an *End-to-End* Neural Network with the purpose of detecting the bite events throughout the recording. During the next step, we process the resulting bite detections using signal processing algorithms to obtain the final meal start and end timestamp estimates. Evaluation results obtained from our Leave One Subject Out experiments using our publicly available FIC and FreeFIC datasets, exhibit encouraging results by achieving an F1/Average Jaccard Index of 0.894/0.804.**

I. Introduction

Automated and objective monitoring of one's eating behavior has the potential of playing an important role both as a prevention mechanism, by tracking everyday dietary habits, as well as a medical tool, by monitoring patient compliance to prescribed dietary goals [1]. Currently, the most commonly used method for monitoring eating behavior is the *food diary*, where subjects report their eating habits. Despite the fact that food diaries are easy to use and provide useful information, most of the time are highly inaccurate [2]. The need for an objective way of monitoring the eating behavior of individuals ignited the interest of the research community and the industry to pursue automated solutions.

Over the past few years, several methods that involve various sensor types have been proposed in the literature. The majority of these methods aim at measuring eating behavior (such as number of bites, eating speed and total food intake) during a meal session. Recent examples are [3] and [4] which involve the usage of wrist-mounted inertial sensors, [5] which uses a weight scale and [6] that makes use of a camera. The downside of these approaches is that require from the user to enable the capturing mechanism prior to every meal and disable it after the end of it.

Automated detection of eating occurrences in free living conditions is a more challenging problem, due to the wide variety of non-eating activities and movements that can lead to false positive detections. It is also a less studied problem.

The work of [7] presents a novel sensory system integrated into a pair of glasses that involves a piezoelectric strain sensor and an accelerometer sensor. In their experiments they achieve an average F1 score of 0.998 when differentiating between food intake and activity level by using two-stage classification scheme involving Support Vector Machines (SVMs) and Decision Trees. A similar work presented in [8], proposes the integration of an Electromyography (EMG) sensor on a pair of eyeglasses towards the detection of meals in free-living environments along with the recognition of food hardness. The authors report a meal detection accuracy of 0.95 in their dataset of 10 subjects.

The authors of [9] explore the feasibility of using the camera of typical smartphone worn around the participants neck as a necklace in order to detect moments of eating in free-living environments. In their dataset of 5 participants providing data over the course of 3 days, the authors report an eating moment recognition accuracy of 0.896.

The work of [10] proposes a solution that uses the accelerometer and gyroscope signals of a novel watch-like device with the aim of classifying periods of everyday life as eating or non-eating. The idea behind their approach is that meals tend to be preceded and succeeded by periods of vigorous wrist motion. In their experimental section the authors report an accuracy of 0.81 for eating detection in a large dataset of 43 subjects.

Motivated by the above, in this paper we propose an approach for detecting meal start and end times in the wild by using the inertial data from a typical smartwatch. The proposed approach is based on our previous work [11] where we presented an end-to-end learning mechanism capable of detecting bite events from *in-meal* data. In this work we extend [11] by incorporating free-living data and we present how bite predictions can lead to the detection of meals. Experimentation using our newly-introduced FreeFIC dataset that contains 16 free-living recordings from 6 subjects yield an encouraging F1 score of 0.894. The dataset is publicly available on the Multimedia Understanding Group site[1].

The rest of the paper is organized as follows. Section II presents the proposed meal detection algorithm. Section III describes the dataset, the experiments and the adopted evaluation scheme. Finally, the paper concludes with the conclusions in Section IV.

II. Meal detection algorithm

A. Signal pre-processing

Consider $\mathbf{a}_x$, $\mathbf{a}_y$, $\mathbf{a}_z$, $\mathbf{g}_x$, $\mathbf{g}_y$, $\mathbf{g}_z$, the synchronized x, y and z accelerometer and gyroscope signals respectively, captured during a single recording. For a single point, indexed by i, $\mathbf{x}(i) = \left[a_x(i), a_y(i), a_z(i), g_x(i), g_y(i), g_z(i)\right]^\top$ contains the instantaneous inertial measurements. The complete recording can be then represented by $\mathbf{r} = \left[\mathbf{x}(1), \ldots, \mathbf{x}(M)\right]$, with $i =$

All authors are with the Multimedia Understanding Group, Information Processing Laboratory, Aristotle University of Thessaloniki, Greece

[1] Available at: `mug.ee.auth.gr/intake-cycle-detection`

$1, \dots, M$. The length of the recording, in samples, is defined as $M = t \cdot f_s$, where t is the recording duration in seconds and f_s the sampling frequency of the sensors in Hz.

We initially smoothed each of the triaxial acceleration and gyroscope streams by convolving them with a moving average filter. Experimentation with a filter length corresponding to 0.25 seconds ($0.25 \cdot f_s$ samples) led to satisfactory results. In addition, since the accelerometer measurements include a component due to the Earth's gravity in addition to voluntary movements, we convolved each of the $\mathbf{a}_x$, $\mathbf{a}_y$ and $\mathbf{a}_z$ streams with a high-pass Finite Impulse Response (FIR) filter with a cut off frequency of 1 Hz and a length corresponding to 5.12 seconds ($5.12 \cdot f_s$ samples).

B. Bite detection

We detect the bite events contained in the raw data series $\mathbf{r}$ using our end-to-end Neural Network (NN) architecture [11] that involves both convolutional and recurrent layers which we summarize here. This end-to-end NN makes use of convolutional layers to extract problem-specific features, as well as a Long-Short Term Memory (LSTM) layer to model the evolution of the extracted features over time. The convolutional part of the NN is comprised of three 1D convolutional layers, with the first two being followed by max pooling operations that decimate the output by a factor of 2. The output of the third convolutional layer is then sequentially processed by the LSTM layer. The final output of the network is obtained using a fully connected layer with a single neuron. Table I summarizes the NN architecture.

TABLE I: End-to-end bite detection architecture [11]. Table provides the numerical parameters and the activation function, where applicable, used in each layer.

Layer	**Dim.**	**Activation**	**Details**
1D Conv	32×5	ReLU	Num of filters × filter len
Max Pool	$2 \downarrow$	–	Decimation factor
1D Conv	64×3	ReLU	Num of filters × filter len
Max Pool	$2 \downarrow$	–	Decimation factor
1D Conv	128×3	ReLU	Num of filters × filter len
LSTM	128	Hard sigmoid	Num of hidden cells
Dense	1	Sigmoid	Num of neurons
# params	$163,617$		

Training examples are obtained by extracting parts of the raw sensor data series $\mathbf{r}$ using a sliding window with length w_l and step w_s. The label paired to each training example corresponds to the label associated with the sample at end of the sliding window. To avoid issues with class imbalance since the negative class is much more frequent, each batch contained an equal amount of positive and negative examples selected at random. The total number of examples in the batch sums to 768. Finally, the network is trained for 4 epochs[2] by minimizing the cross-entropy loss using the RMSprop optimizer and a learning rate of 10^{-3}.

Bite detection is initially carried out by processing a previously unseen recording $\mathbf{r}$ using the NN to obtain the predictions $\mathbf{p}$ of length $N = \frac{M}{4}$ (decimated by a factor of 4 due to the pooling operations, additional details about inference can be found in [11]). We then perform thresholding in $\mathbf{p}$ by replacing with zeros the elements that are below a probability threshold λ_p. By performing a local maxima search in $\mathbf{p}$, using a minimum distance of 2 seconds between consecutive peaks, we obtain the set of detected bites $\mathcal{B} = \{b_1, \dots, b_L\}$, where each b_l with $l = 1, \dots, L$, corresponds to the timestamp associated with a local maximum.

C. Meal detection

Given the set of detected bites $\mathcal{B}$ for a recording $\mathbf{r}$, estimation of meal start and end times starts by constructing the timeseries $\mathbf{s}(n)$ with $n = 1, \dots, N$, as in Equation 1.

$$\mathbf{s}(n) = \begin{cases} 1, & \text{if } n = b \cdot \frac{f_s}{4} \quad \forall\, b \in \mathcal{B} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Essentially, $\mathbf{s}$ spans the entire recording duration and is positive at the moments of bite detections and zero elsewhere. Next, $\mathbf{s}$ is convolved with a *Gaussian* filter with a length and σ that correspond to 240 and 45 seconds ($\frac{240 \cdot f_s}{4}$ and $\frac{45 \cdot fs}{4}$ samples), respectively. This process smooths $\mathbf{s}$ and closes the gaps (similar to the morphological *closing* operation) between groups of bites that are distant from each other, which is often the case in long meals. The timeseries $\mathbf{s}$ is then thresholded by replacing with zeros the elements that are lower than a threshold λ_s and with ones the elements that surpass it.

Subsequently, we convolve the binary $\mathbf{s}$ timeseries with the differentiation filter $\mathbf{h}$ to obtain the N-length series $\mathbf{d}$. Filter $\mathbf{h}$ is selected to have a sidelobe length corresponding to 1 second ($\frac{f_s}{4}$ samples) and is constructed as: $\mathbf{h} = [1, 2, \dots, \frac{f_s}{4}, 0, -\frac{f_s}{4}, \dots, -2, -1]$.

An initial estimate of the meal start and end times is achieved by performing a local maxima search in $|\mathbf{d}|$ and then pairing consecutive peaks to isolate stable regions. As a result of this process we obtain the set of initial meal intervals $\mathcal{Q} = \{q_1, \dots, q_V\}$ where each interval $q_i = [t_i^l, t_i^r]$ contains the left-most and right-most edges between stable regions, t_i^l and t_i^r respectively, with $i = 1, \dots, V$. In addition we use the bite estimates from $\mathcal{B}$ to discard the intervals q_i that include less than 3 bites in their duration. The final estimate of the meal start and end times is obtained by iteratively merging the elements of $\mathcal{Q}$ that are within 180 seconds of each other. Figure 1 presents the steps of the meal detection process. Looking at Figure 1-e) one can consider that a subsequent "short meal" rejection procedure would reduce false detections.

III. Experiments & Evaluation

A. Dataset

In this work we make use of two datasets. The first is our publicly available *Food Intake Cycle* (FIC) dataset that includes 21 recordings from *within-meal* sessions belonging to 12 subjects. The second is the newly introduced *FreeFIC* dataset that includes 16 *in the wild* recordings belonging to

[2] An epoch ends when all *negative* examples are used once in the optimization process.

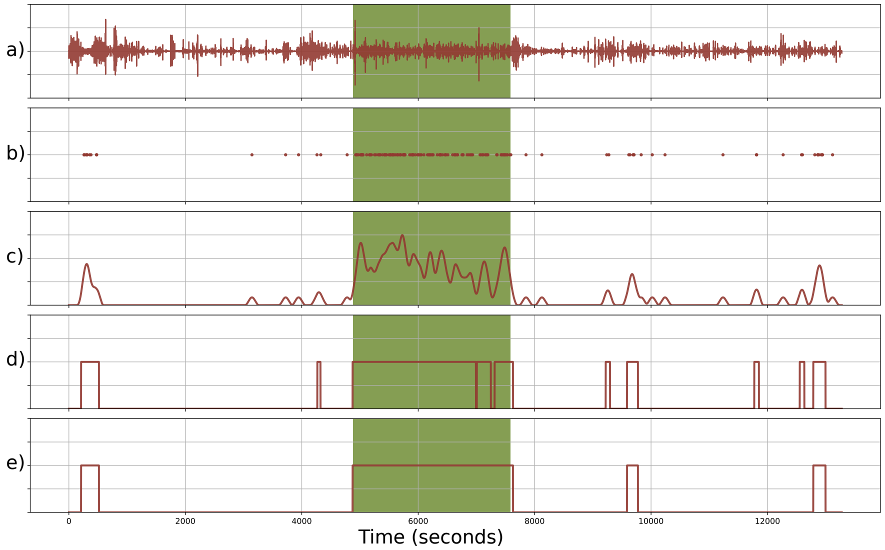

Fig. 1: Figure depicting the steps of the meal detection algorithm. In all figures the horizontal axis is aligned and spans the entire recording duration, as well as the *true* meal interval is marked with a green background. In detail: a) shows the raw $\mathbf{a}_x$ stream, b) the dot markers represent the detected bites (set $\mathcal{B}$) made by the NN (Section II-B), c) shows the application of the Gaussian filter and the resulting $\mathbf{s}$ timeseries, d) and e) show the initial and the final meal estimates (set $\mathcal{Q}$), respectively.

TABLE II: FIC and FreeFIC dataset statistics.

Dataset	**Type**	**#**	**Mean (s)**	**Std (s)**	**Median (s)**	**Total (s)**
FIC	Meal sessions	21	703.56	186.18	717.88	$14,774.80$
	Food Intake Cycles	$1,332$	4.52	3.22	3.55	$6,023.07$
FreeFIC	In the wild sessions	16	$17,398$	$4,884$	$16,489$	$278,378$
	Meals	17	$1,148$	502	$1,065$	$19,520$

6 subjects. In contrast to FIC sessions where the average recording duration is 703 seconds, FreeFIC recordings span a significantly larger portion of the day ($17,398$ seconds on average) including free-living activities and contain at least one meal. In both datasets the recorded data consists of tri-axial accelerometer and gyroscope measurements originating from a commercial smartwatch.

Participants in FreeFIC were advised to wear the smartwatch, to the wrist that they typically use to operate the fork and the spoon, well-ahead before having their meal and continue wearing it afterwards until the smartwatch reached critical battery levels. Apart from noting the start and end moments of their meals to the best of their abilities (with less than a minute resolution), no other instructions were given to the participants.

Annotations in FIC include the start and end moments of each food intake cycle (i.e. bite event) during the meal, while annotations in FreeFIC include the start and end moments of meals throughout the day. For the raw signal recordings $\mathbf{r}$ in the FIC dataset we associated each $\mathbf{x}(i)$ within 0.1 of the end of a food intake cycle interval with a positive label, all other samples are assigned labels from the negative class. Regarding the raw signal recordings $\mathbf{r}$ from the FreeFIC dataset, we associated each $\mathbf{x}(i)$ *outside* of the meal intervals with a label from the negative class. Table II presents the statistics of the FIC and FreeFIC datasets.

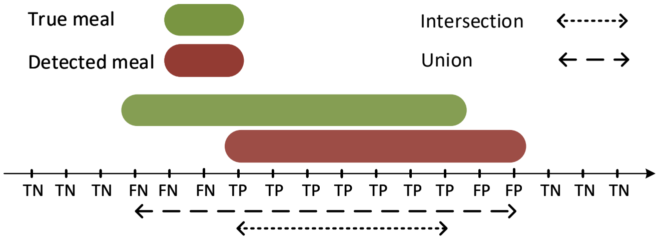

Fig. 2: Example presenting the calculation of performance metrics using the adopted evaluation scheme.

B. Experiments & evaluation methodology

For our experiments, we extracted positive and negative examples from the FIC dataset using a w_l and w_s that correspond to 5 and 0.05 seconds ($5 \cdot f_s$ and $0.05 \cdot f_s$ samples), respectively. We also extracted negative examples from the FreeFIC dataset using a less exhaustive, due to FreeFIC recordings being significantly larger in duration, w_s' equal to 1 second ($1 \cdot f_s$ samples). Furthermore, we resampled recordings in both datasets to a constant sampling frequency $f_s = 100$ Hz. We set the probability threshold λ_p (from Section II-B) to 0.89 according to [3]. Finally, experimenting with a small part of the FreeFIC dataset (four out of the

TABLE III: Leave One Subject Out meal detection results.

Method	TP	FP	TN	FN	Precision	Recall	Specificity	F1	Average $\mathcal{J}$ Index
Proposed	$432,917$	$47,187$	$6,424,247$	$55,083$	**0.901**	0.887	**0.992**	**0.894**	**0.804**
DBSCAN	$436,955$	$84,460$	$6,386,974$	$51,045$	0.838	**0.895**	0.986	0.865	0.752

sixteen recordings selected at random) we set threshold λ_s (from Section II-C) to $5 \cdot 10^{-4}$ as this value achieved the highest F1 score for that small subset.

The purpose of our experiments is to measure the inter-subject effectiveness of our approach. To this end, we trained the end-to-end bite detection network using data from FIC and FreeFIC in a Leave One Subject Out (LOSO) fashion by iteratively leaving out recordings from both datasets belonging to each unique subject. Each model was then used to produce the bite estimates for the recordings of the left out subject and forward them to the proposed meal detection algorithm.

In addition to the meal detection experiment, we performed a LOSO experiment investigating the differences in *bite detection* performance when including recordings from both datasets (FIC and FreeFIC) in the network's training process against only recordings from the FIC dataset. Evaluation is performed on the left out meal sessions from the subjects in FIC.

For comparison purposes we evaluate the performance of the state-of-the-art *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) algorithm towards the detection of meals in FreeFIC. This is motivated by [4] where the authors use density-based clustering to detect the final bite events from a continuous stream of bite decisions. Meal detection using DBSCAN is performed by providing as input the timestamps of the detected bites (set $\mathcal{B}$, Section II-B). To be in par with the proposed approach and achieve a fair comparison we tuned the clustering algorithm according to the proposed one (Section II-C; specifically, we used a *minimum number of samples* equal 3 (bites) and a *maximum distance between samples* (or *eps*) of 180 seconds. The final meal estimates are obtained by pairing the extremums of each produced cluster.

By considering the complete recording timeline, as well as that each point in time belonging to a q_i interval corresponds to the positive class (i.e. meal) while the rest correspond to the negative class (i.e. non-meal), we can exhaustively calculate the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) metrics. Moreover, we measured the overlap of the estimated meals against the true meal intervals using the *Jaccard Index* defined as $\mathcal{J}(Q,T) = \frac{|Q \cap T|}{|Q \cup T|}$, where Q and T are the estimated and true meal intervals respectively. An example of how metric calculation is performed is depicted in Figure 2.

C. Results & discussion

Table III presents the LOSO meal detection results obtained by the proposed and the DBSCAN approaches. The results initially point out that the presented approach outperforms the density-based clustering approach by achieving an F1/Average $\mathcal{J}$ Index of $0.894/0.804$ against $0.865/0.752$. This difference in performance stems from the observation that DBSCAN produces almost twice the amount of FPs despite yielding slightly more TPs and slightly less FNs than the proposed approach. Regarding the cross-subject bite detection performance, introducing examples from both FIC and FreeFIC datasets in the network's training process resulted in a similar performance when compared to using examples solely from the FIC dataset, with F1 scores of 0.872 and 0.884 respectively.

IV. Conclusions

In this work we presented an algorithm for detecting meals in the wild using the inertial data (acceleration and orientation velocity) of a typical smartwatch. Evaluation is performed in a LOSO fashion using our newly introduced and publicly available FreeFIC dataset, where the proposed approach yields an F1/average Jaccard Index of $0.894/0.804$.

V. Acknowledgments

The work leading to these results has received funding from the EU Commission under Grant Agreement No. 727688 (`http://bigoprogram.eu`).

References

[1] D. B. Sarwer *et al.*, "Preoperative eating behavior, postoperative dietary adherence, and weight loss after gastric bypass surgery," *Surgery for Obesity and Related Diseases*, vol. 4, no. 5, pp. 640 – 646, 2008.

[2] D. A. Schoeller, "How accurate is self-reported dietary energy intake?" *Nutrition reviews*, vol. 48, no. 10, pp. 373–379, 1990.

[3] K. Kyritsis, C. Diou, and A. Delopoulos, "Modeling wrist micromovements to measure in-meal eating behavior from inertial sensor data," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2019.

[4] S. Zhang *et al.*, "Food watch: Detecting and characterizing eating episodes through feeding gestures," in *11th EAI International Conference on Body Area Networks*, ser. BodyNets '16, 2016, pp. 91–96.

[5] V. Papapanagiotou, C. Diou, I. Ioakimidis, P. Sodersten, and A. Delopoulos, "Automatic analysis of food intake and meal microstructure based on continuous weight measurements," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2018.

[6] M. M. Anthimopoulos *et al.*, "A food recognition system for diabetic patients based on an optimized bag-of-features model," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, 2014.

[7] M. Farooq and E. Sazonov, "A novel wearable device for food intake and physical activity recognition," *Sensors*, vol. 16, p. 1067, 2016.

[8] R. Zhang and O. Amft, "Monitoring chewing and eating in free-living using smart eyeglasses," *IEEE journal of biomedical and health informatics*, vol. 22, no. 1, pp. 23–32, 2018.

[9] 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*. ACM, 2013, pp. 26–33.

[10] Y. Dong *et al.*, "Detecting periods of eating during free-living by tracking wrist motion," *IEEE journal of biomedical and health informatics*, vol. 18, no. 4, pp. 1253–1260, 2014.

[11] K. Kyritsis, C. Diou, and A. Delopoulos, "End-to-end learning for measuring in-meal eating behavior from a smartwatch," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, July 2018, pp. 5511–5514.