End-to-end Learning for Measuring in-meal Eating Behavior from a Smartwatch

Konstantinos Kyritsis, Christos Diou and Anastasios Delopoulos

***Abstract*— In this paper, we propose an *end-to-end* neural network (NN) architecture for detecting in-meal eating events (i.e., bites), using only a commercially available smartwatch. Our method combines convolutional and recurrent networks and is able to simultaneously learn intermediate data representations related to hand movements, as well as sequences of these movements that appear during eating. A promising F-score of 0.884 is achieved for detecting bites on a publicly available dataset with 10 subjects.**

I. Introduction

Obesity is the result of imbalance between energy intake and energy expenditure [1]. While the use of wearable devices for measuring energy expenditure in terms of physical activity is now common, the usage of wearable devices for measuring eating behavior is less widespread. In most settings (clinical or otherwise), eating behavior is monitored through self-reported food diaries, which can be highly inaccurate [2]. In addition, there are relevant meal parameters which cannot be provided through self-reports and therefore must be measured, such as eating rate or number of bites during the course of a meal [3].

Numerous methods using different types and number of sensors have been proposed to objectively monitor eating behavior. Examples include weight scales [4], visual sensors [5], the combination of audio and motion sensors [6] [7] or the combination of multiple motion and gesture sensors [8]. Some of these methods automatically identify eating occurrences (meals or snacks), while others measure in-meal eating behavior. They achieve high effectiveness by combining multiple and/or sophisticated sensors, however in some cases this use of additional devices or sensors can reduce usability and hurt compliance.

Approaches that make use of the inertial sensors of a single smartwatch already worn by the user can potentially be easier to use. The work presented in [9] proposes a combination of spectral segmentation, aggregation and Random Forest classification to detect eating episodes with an average F-score of 0.75. The authors of [10] present a method that uses Hidden Markov Models (HMMs) to capture the temporal dependencies of hand gestures leading to a bite. Their method achieves an accuracy of 0.965 when evaluating pre-segmented sequences; no results are reported for detecting bites in a continuous fashion.

In our previous work [11] we characterized the *food intake cycle* as a sequence of specific hand micromovements (e.g. pick food movement, food to mouth movement) and proposed a two-step approach for detecting eating events during a meal by means of a single smartwatch. More specifically, we represented each micromovement as a score vector of 10 Support Vector Machines (SVM), and then used a Long-Short Term Memory (LSTM) network to capture the temporal evolution of the micromovements. The LSTM output was used to identify whether any given sequence is an intake cycle or not, and achieved a Leave-One-Subject-Out (LOSO) F-score of 0.892 in a dataset of 10 subjects. The main drawback of this method is the need for detailed information about hand micromovements for training the SVM detectors of the first step, which requires significant annotation effort.

Motivated by the recent success of end-to-end learning on timeseries data, this paper explores the possibility of using a neural network architecture for detecting eating events during a meal by using the raw accelerometer and gyroscope measurements of an off-the-shelf smartwatch. In the recent years, end-to-end learning approaches that jointly train the convolutional and recurrent parts of Deep Neural Networks (DNNs) have been successfully applied in practical recognition problems such as multi-sensor wearable activity recognition [12] and audio-based emotion recognition [13]. Despite the obvious benefit of the Convolutional Neural Network's (CNN) ability to generate problem-specific data representations, the authors of [12] point out the advantages of using LSTM cells to capture the temporal dependencies of the convolutional activations in contrast to an approach that is based solely on CNNs.

Based on these observations, in our experiments we explored both the use of CNN-only architectures and the combination of CNNs and LSTMs. Evaluation is performed on our publicly available *Food Intake Cycle* (FIC)[1] dataset, and the results show that the combination of CNN with LSTM significantly outperforms CNN-only architectures. Furthermore, the proposed end-to-end-approach achieves similar performance to our previous, state-of-the-art method [11], without the need for micromovement annotations during training.

The rest of this paper is organized as follows. Section II provides the details of the processing and learning pipeline. Section III describes the dataset, the conducted experiments and their results. Finally, Section IV concludes the paper.

II. End-to-end learning of eating events

The term *end-to-end* is used to describe a learning machine that, when given raw information, is able to extract problem-

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

[1] http://mug.ee.auth.gr/intake-cycle-detection

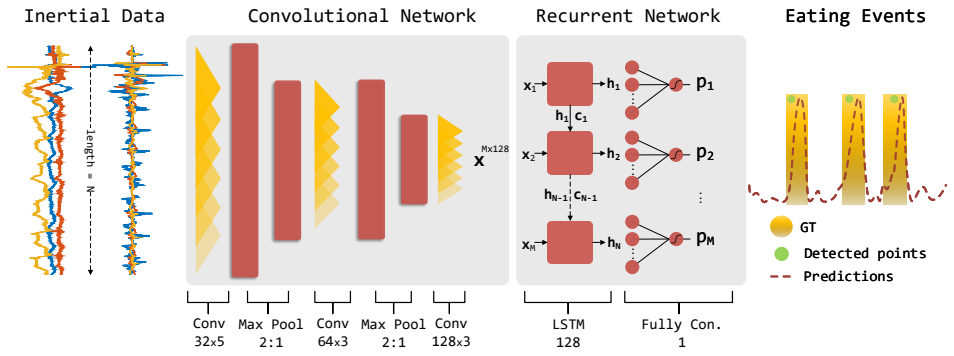

Fig. 1: Pipeline of the proposed approach. Initially, the accelerometer and gyroscope signals of length N are processed by the convolutional part of the architecture producing the $M \times 128$ intermediate output $\boldsymbol{x}$, where $M = \frac{N}{4}$ due to the temporal max pooling operations. Then, the recurrent part of the network produces a probability p_i for each of the M time slices of $\boldsymbol{x}$. The LSTM's output and the cell state for each of the M moments are represented by h_i and c_i respectively.

specific features, as well as to model the evolution of the extracted features across time. Instead of making use of the explicit knowledge of hand micromovements leading to an eating event, the work presented here combines Convolutional and LSTM networks into a single end-to-end learning mechanism. Figure 1 presents the overall pipeline.

A. Data pre-processing

Let a_x^n, a_y^n, a_z^n and g_x^n, g_y^n, g_z^n, with $n = 1, \ldots, N$ be the synchronized x, y and z streams of the accelerometer and gyroscope measurements captured during a meal session. N is defined as $N = t \cdot fs$, where t is the meal's duration in sec and fs the sensors' sampling frequency in Hz. A meal can be then represented as an $N \times 6$ data matrix $\boldsymbol{S}$ defined as $\boldsymbol{S} = [a_x^T\ a_y^T\ a_z^T\ g_x^T\ g_y^T\ g_z^T]$ and a $N \times 1$ vector $\boldsymbol{l}$ indicating the label for every sample.

Initially, we smooth each sensor stream by applying a 5-th order median filter to each column of $\boldsymbol{S}$. Then, a high pass FIR filter with a 512 tap-delay line and a cut-off frequency of 1 Hz is individually convolved with the a_x^n, a_y^n and a_z^n components of $\boldsymbol{S}$ in order to attenuate the acceleration components caused by the earth's gravitational field. Prior to any further processing, $\boldsymbol{S}$ is column-wise standardized by subtracting the mean and dividing by the standard deviation. The result of this process is the $N \times 6$ matrix $\boldsymbol{S'}$.

B. End-to-end network architecture

The proposed end-to-end architecture consists of two networks, a convolutional and a recurrent, that are jointly trained by back-propagation using the same cost function. The purpose of the convolutional network is to extract data representations related with short-duration movements of the hand; the recurrent network models the evolution of hand movements leading to an eating event.

The CNN is comprised of three 1-D convolutional layers; the first two being followed by a *temporal max pooling* operation with a decimation factor of 2. As the depth of the network increases, the number of filters in each convolutional layer increases as well. More specifically we set 32, 64 and 128 as the number of filters for each layer. The filters' length were set to $\frac{5}{fs}$, $\frac{3}{\frac{fs}{2}}$ and $\frac{3}{\frac{fs}{4}}$ sec. The Rectified Linear Unit (ReLU) was used as the non-linearity for the convolutional activations.

The recurrent network consists of a single LSTM layer with 128 cells; however similar performance was obtained with two LSTM layers of 128 cells each (see the experiments of Section III). The hard sigmoid function defined as $\sigma(x) = \max(0, \min(1, x \cdot 0.2 + 0.5))$ was selected as the activation of the recurrent steps. During training the output of the network is obtained by processing the *last* 128-dimensional output of the LSTM with a fully connected layer with a single neuron and the sigmoid function as the non-linearity. The output of the network represents the probability that the given sequence ends with an eating event. As a form of regularization we applied a 20% dropout chance on the LSTM inputs, a 50% dropout chance on the units of the LSTM state during the recursion steps and a 50% dropout chance to the inputs of the fully connected layer.

For training examples we extracted parts of $\boldsymbol{S'}$ (Section II-A) using the sliding window approach. In more detail, the length w_l and the step w_s of the sliding window in samples are defined as $w_l = 5 \cdot fs$ and $w_s = 0.05 \cdot fs$ (5 and 0.05 sec respectively). The label associated with each section corresponds to the label at the end of the extraction window. This process results in a collection of $w_l \times 6$ sequences, each coupled with a single label as the target. We trained the network using the binary cross-entropy cost function coupled with the RMSprop optimizer with a learning rate of 10^{-3} for 16 epochs and a mini-batch size of 128 sequences.

During inference we modified the recurrent network in order to fully benefit from the LSTM's memory regarding past inputs. More specifically, the LSTM layer is modified to propagate *all* intermediate outputs (not just the last, in contrast with the training network) to the fully connected layer. Additionally, the fully connected layer is modified to provide a prediction p for each of the intermediate outputs of the LSTM layer. This means that when an $N \times 6$ data frame representing a complete meal is fed to the evaluation network, the state of the LSTM is propagated throughout the entire meal and the network outputs a $\frac{N}{4} \times 1$ (downsampled

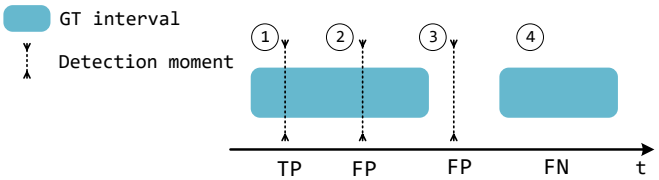

Fig. 2: The figure depicts four representatives examples of the evaluation scheme used to calculate performance metrics.

by 4 due to temporal max pooling) prediction vector $\boldsymbol{p}$.

C. Detection

Given the trained network and a data matrix $\boldsymbol{S'}$ that represents a meal session, eating events are detected by processing the network's predictions $\boldsymbol{p}$ using the following steps. Initially, we use a 5-th order median filter to smooth any short and sudden changes in $\boldsymbol{p}$. The sequence $\boldsymbol{y}$ is then obtained by convolving $\boldsymbol{p}$ with an one-dimensional edge detector $\boldsymbol{h}$, defined as $\boldsymbol{h} = [1, 2, 3, 0, -3, -2, -1]^T$. We then normalize $\boldsymbol{y}$ in $[0, 1]$ and the $\boldsymbol{y'}$ is subsequently formed by replacing with zeros the elements of $\boldsymbol{y}$ that are below a threshold T_y. Finally, eating events are detected by performing a local maxima search in $\boldsymbol{y'}$ with a minimum distance between peaks set at 1 sec.

III. Experiments & Results

A. Dataset

The data used in this study belong in our publicly available FIC dataset. The dataset contains Inertial Measurement Unit (IMU) recordings of 10 meal sessions originating from 10 unique subjects using a Microsoft Band 2 smartwatch, in the restaurant of Aristotle University of Thessaloniki (Figure 3). Both the accelerometer and gyroscope streams are synchronized and resampled at a sampling frequency fs equal to 100 Hz. FIC also provides detailed annotations regarding the start and end moments of specific non-overlapping hand micromovements that span the entire length of a meal. More specifically, the micromovements contained in FIC are: *i) pick food* (hand manipulating utensil to pick food from plate), *ii) upwards* (upward motion of the hand towards the mouth), *iii) mouth* (hand inserts food in mouth), *iv) downwards* (downwards motion of the hand, away from the mouth), *v) no movement* and *vi) other movement* for describing any micromovement not belonging in any of the aforementioned ones. Additional technical details about FIC can be found in the provided hyperlink.

FIC's micromovement annotations are not used in our end-to-end setup, but we adopt the definition of the intake cycle as in [14] (i.e. as a sequence of hand micromovements starting with the *pick food*, ending with *downwards* and containing a *mouth* micromovement). We populated each meal's label vector $\boldsymbol{l}$ with positive values only for 0.1 sec before and after the end of an intake cycle, and negative values everywhere else. More formally,

$$l^n = \begin{cases} +1 & if \quad t^i_{gt} - 0.1 \leq l^n_t \leq t^i_{gt} + 0.1 \\ -1, & else \end{cases}$$

Fig. 3: Stills from the video sequences used to annotate the FIC dataset. The figure reflects the various eating styles and food types in the dataset.

where t^i_{gt} is the timestamp at the end of the i^{th} intake cycle in sec and l^n_t the timestamp associated with the n^{th} element of $\boldsymbol{l}$.

B. Evaluation Scheme

Given the detected eating events and the ground truth intervals of a meal session, the method's performance is evaluated by calculating the true positive (TP), false positive (FP) and false negative (FN) metrics as in [11]. An overview of the evaluation scheme is presented in Figure 2. Fundamentally, the first detection point in a ground truth interval is consider as a TP, whereas any excess detection points in the same interval count as FPs. Additionally, detection points not belonging in any interval also count as FPs. Finally, ground truth intervals with no detection points count as FNs. In the context of this study we allowed any detections within a tolerance of 0.5 sec past the end of an interval to count as TPs (given that no other detections for this interval exist). Furthermore, unlike [10] where the authors evaluated the performance of their approach against pre-segmented sequences, in this work evaluation is carried out against continuous (i.e. non-segmented) sequences.

C. Experiments

As a baseline for comparing our method we used our previous works of [14] and [11], both making use of the explicit knowledge of each of the micromovements leading to an eating event. The former work (M-I) uses two HMMs to model the *decisions* of a multiclass SVM and make predictions regarding whether or not a given sequence is an intake event or not. The short description of the latter (M-II) approach is already provided in Section I.

We also compare the proposed network architecture with different NN architectures that don't make use of the knowledge regarding micromovements. Initially, we compare the performance against an architecture that only uses convolutional and fully connected layers and approximately has the same number of learnable parameters. The convolutional-only (A-I) architecture consists of 5 convolutional layers and 1 fully connected layer with a single neuron. On the same

TABLE I: Micromovement agnostic architectures. The notation *Conv* ($N \times M$) indicates a 1D convolutional layer with N filters each with a length of M samples, *Pool* ($K \downarrow$) indicates the max pooling operation with a decimation factor of K, *LSTM* (H) indicates an Long-Short Term Memory layer with H hidden cells and *Dense* (Q) represents a fully connected layer with Q neurons.

ID	Proposed	A-I	A-II
Layers	Conv (32×5) Pool ($2 \downarrow$) Conv (64×3) Pool ($2 \downarrow$) Conv (128×3) LSTM(128) Dense (1)	Conv (32×20) Conv (32×20) Pool ($2 \downarrow$) Conv (64×10) Conv (64×10) Pool ($2 \downarrow$) Conv (128×4) Dense(1)	Conv (32×5) Conv (32×5) Pool ($2 \downarrow$) Conv (64×3) Conv (64×3) Pool ($2 \downarrow$) Conv (128×3) LSTM(128) LSTM(128) Dense (1)
# parameters	163,617	134,849	312,705

TABLE II: Evaluation results. The *micromovement* column indicates whether or not the approach uses knowledge from hand micromovements.

micromovements	**ID**	**Prec**	**Rec**	**F-score**
✗	Proposed	0.852	0.924	0.884
✗	A-II	0.824	0.860	0.839
✗	A-I	0.675	0.699	0.676
✓	M-II	0.875	0.911	0.892
✓	M-I	0.757	0.881	0.814

note as the proposed architecture, A-I includes two temporal max pooling operations after the second and the fourth convolutional layers. Finally, we compare the proposed network with a deeper version (A-II), that is the same as the A-I architecture with the exception of two LSTM layers prior to the fully connected layer. In order to avoid overfitting, both in A-I and in A-II the fully connected layer discards activations from the previous layer with a probability of 50%. Table I aggregates the information regarding all micromovement agnostic neural network architectures. In order to measure the cross-subject performance, all experiments were conducted using the LOSO cross validation scheme. By experimenting with a small part of the dataset, the T_y parameter was selected to be 0.4. Table II presents the precision, recall and F-score metrics calculated according to the evaluation scheme of Section III-B. The presented results are averaged across all LOSO iterations.

Results show that the CNN-only architecture (A-I) underperforms when compared with either micromovement-based or micromovement-agnostic approaches, indicating the need for modeling of time-evolving and variable in length phenomena that occur during eating. Additionally, the performance of both the proposed and the deeper A-II architectures surpass the M-I approach; however only the shallower proposed architecture achieves almost identical state-of-the-art performance as M-II.

IV. Conclusions

In this paper we presented an end-to-end neural network approach for detecting eating events during the course of meal using raw data from the accelerometer and gyroscope sensors commonly found in commercially available smartwatches. In addition, experimental results show that the proposed end-to-end approach yields state-of-the-art performance, despite not having knowledge of hand micromovements leading to an eating event during training.

V. Acknowledgments

The work leading to these results has received funding from the EU Commission under Grant Agreement No. 727688 (`http://bigoprogram.eu`, H2020). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

[1] W. H. Organization, *Obesity: preventing and managing the global epidemic*. World Health Organization, 2000, no. 894.

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

[3] C. Maramis, C. Diou, I. Ioakeimidis, I. Lekka, G. Dudnik, M. Mars, N. Maglaveras, C. Bergh, and A. Delopoulos, "Preventing obesity and eating disorders through behavioural modifications: the SPLENDID vision," in *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on*, 2014, pp. 7–10.

[4] V. Papapanagiotou, C. Diou, B. Langlet, I. Ioakimidis, and A. Delopoulos, "A parametric probabilistic context-free grammar for food intake analysis based on continuous meal weight measurements," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, 2015, pp. 7853–7856.

[5] 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.

[6] 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*, vol. 21, no. 3, pp. 607–618, 2017.

[7] M. Mirtchouk *et al.*, "Automated estimation of food type and amount consumed from body-worn audio and motion sensors," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 451–462.

[8] A. Doulah *et al.*, "Meal microstructure characterization from sensor-based food intake detection," *Frontiers in nutrition*, vol. 4, p. 31, 2017.

[9] S. Zhang, W. Stogin, and N. Alshurafa, "I sense overeating: Motif-based machine learning framework to detect overeating using wrist-worn sensing," *Information Fusion*, vol. 41, pp. 37–47, 2018.

[10] R. I. Ramos-Garcia *et al.*, "Improving the recognition of eating gestures using intergesture sequential dependencies," *IEEE journal of biomedical and health informatics*, vol. 19, no. 3, pp. 825–831, 2015.

[11] K. Kyritsis, C. Diou, and A. Delopoulos, "Food intake detection from inertial sensors using lstm networks," in *International Conference on Image Analysis and Processing*. Springer, 2017, pp. 411–418.

[12] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[13] G. Trigeorgis *et al.*, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2016, pp. 5200–5204.

[14] K. Kyritsis, C. L. Tatli, C. Diou, and A. Delopoulos, "Automated analysis of in meal eating behavior using a commercial wristband imu sensor," in *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*, 2017, pp. 2843–2846.