FIT-EVE&ADAM: Estimation of Velocity & Energy for Automated Diet Activity Monitoring*

Junghyo Lee, Prajwal Paudyal, Ayan Banerjee, and Sandeep K. S. Gupta
iMPACT Lab, CIDSE, Arizona State University
Email: {jlee375, ppaudyal, abanerj3, and sandeep.gupta}@asu.edu

***Abstract*—State-of-the-art techniques for eating activities analysis in dietary monitoring require significant user intervention, which is reported to be one of the major reasons for low adherence. There are limited works using wearables for fine-grained analysis of eating activities in terms of the eating speed, the type of food consumed, and the portion sizes. In this paper, we propose FIT-EVE&ADAM, an armband based diet monitoring system that provides such fine-grained analysis, triggered by a single hand gesture. The system collects the user's gesture using sensors such as electromyogram embedded in the armband device, along with food image data using color and thermal cameras. Finally, a novel feature selection method is applied on the data features to estimate eating speed and caloric intake with high accuracy (0.96 F1 score).**

***Keywords*-Diet Monitoring; Gesture Recognition; Wearable**

I. Introduction

Dietary monitoring has been considered to be an important aspect of treatment plans for many common health problems [1]. The primary problem that plagues nearly all diet monitoring solutions is low adherence, which is caused mainly by the requirement of manual input from users. A diet monitoring system typically includes *food identification*, and *eating activity monitoring* that can together determine utensils used, plate section information, portion sizes for each morsel, and finally the instantaneous eating speed. All this information is vital to be able to accurately compute total calories consumed in any given meal as well as to provide useful feedback to the user regarding her eating speed (Fig. 1). However, gathering this information with high levels of accuracy is extremely challenging, especially with minimal manual intervention. An automatic solution to this problem requires a smart system that can recognize patterns related to eating activities and distinguish them from non-eating behaviors. Such a solution also needs to be adaptive to the behavior of user, to a variety of eating speeds and food-types. The speed of eating can be computed by observing each eatings cycles during a meal. Real-time feedback can then be given if the speed at any time exceeds a set threshold. Calculation of the total calorie intake however is a more involved process, which requires summing up the total calories consumed per eating cycle. An eating cycle consists of a user picking up some food, carrying the food to the mouth, and putting it in the mouth. For calculating the total caloric intake, information such as the *food type, number of times user consumed particular type of food, and weight of each food type* is needed. In this sense, $Total\ Calories = \sum_{i=1}^{N} CC_i W_i$, where CC_i is the calorie content per unit weight and W_i is each morsel weight for food type i.

*This work has been partly funded by CNS grant #1218505, IIS grant #1116385, and NIH grant #EB019202.

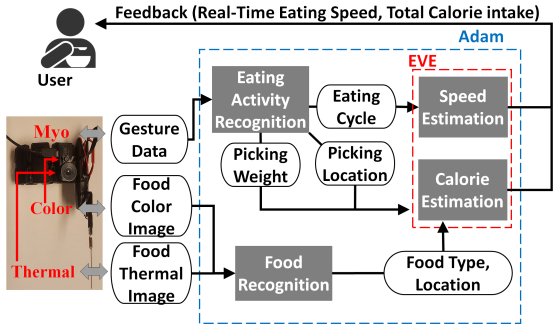

Figure 1: FIT-EVE&ADAM System Model; white boxes represent data blocks and grey boxes represent processes.

Automatically determining the number of eating cycles is a non-trivial problem since it requires detection of the exact times for each eating cycle. There are several challenges some which are as follows: 1) de-noising challenge: during a meal the users not only perform eating actions but also may move their hands in random patterns for example during a conversation. During this episode, the armband sensor datasets are continuously gathered. De-noising the data and detecting meaningful actions such as the user picking up food and putting food in her mouth, is a considerably difficult task, 2) data imbalance challenge: the second challenge is that as the meaningful actions are very transient, the amount of data for picking and eating actions are small compared to the non-meaningful actions such as random hand movements that do not pertain to the eating cycle, and 3) real time challenge: identification of correct eating cycles requires supervised learning with a large combination of features from accelerometer, gyroscope, orientation, and EMG. Furthermore, there can be a lot of possible candidate start and end times for an eating cycle. Finding the correct start and end time among many such candidates is also computationally expensive.

The weight of each morsel can be determined by sensing subtle variations in muscle tension when a user picks up some food. Electromyogram (EMG) can be used to detect such variations in muscle tensions. Although EMG activity is highly variable over time and dependent on the individual, our experiments show that for a given individual, EMG readings at certain parts of the arm show significant difference in

activation when different weights are picked from the plate.

The main contribution of this paper is the *FIT-EVE&ADAM* armband system to monitor eating activity, which has two modules: 1) Eating time and speed estimation and 2) Calorie estimation. Calories estimation includes the following submodules: a) plate section identification: application of supervised learning to identify the plate section from which the user picks food and b) estimation of portion size in each morsel: application of *Dynamic Time Warping (DTW)* using EMG signals to compute food weight picked by a user.

II. Related work

State-of-the-art smartphone camera based monitoring systems [2, 3, 4] have been suggested to solve the problems associated with self-reporting techniques. However, the reported accuracy of such techniques are only around 70% for identification of cooked food. Our previous work, MT-Diet [5], fuses color and thermal images of cooked food to obtain an accuracy of nearly 90%. However, the technique requires placement of two water-filled caps in camera view, which can be an unreasonable requirement for the user. Moreover, these systems still depend on the user having to report the actual portion of food consumed or take a follow-up picture after the meal, or put a finger in the picture frame to determine exact portion size, which are not usable and a cause for termination of monitoring for several users.

Recently crowd-sourcing have been proposed to identify food items but it requires inputs from a large set of users and has privacy concerns [6]. Systems based on camera [2, 3, 4] either estimate calories based on only one image, or require the user to take before-and-after images. The estimation of actual food consumed is based on 2D pixel-by-pixel difference which has low accuracy for volume estimation.

Another set of works use commodity and/or specialized sensors for activity monitoring [7, 8]. Using necklace or ear-wearable systems to monitor eating activity are novel ideas, however the systems proposed are either unable to identify type of food at all or are susceptible to variations in sitting positions of user. A distracted eating pattern is when a user is involved in other activities like talking, swallowing saliva, moving around, and multiple picking before eating. None of the works mentioned above deal with such scenarios, which is expected in real life situations.

III. Methodology

Fig. 1 shows the overall operation of the *FIT-EVE&ADAM* system. The system has two main components a) food recognition and b) eating activity recognition. For the food recognition, we used our preliminary system, MT-Diet [5]. Eating activity recognition has three sub-components, a) eating cycle, b) plate section which user picked, and c) food weight for each morsel.

As seen Fig. 1, our armband based prototype including the color and thermal camera is interfaced to smartphone. *FIT-EVE&ADAM* is triggered by predefined user hand gesture. Then, the two main components work in parallel. In food recognition, the prototype takes a color and thermal image and sends it the cloud server to identify the food type and location, and estimate calories. In the eating activity recognition, counts the number of eating actions which can be used to measure the eating speed.

A. Eating Cycles Detection

The eating cycle consists of three sub-actions: 1) user picking up food, 2) carrying food to the mouth, and 3) putting food in the mouth. All other actions during a meal are considered to belong to 'other' class. Eating cycle detection has the following sub-tasks:

1) Segmentation: Since eating cycle duration is not uniform, the sliding windows with fixed window size had low accuracy. To overcome this obstacle, the sliding window with all possible window sizes are considered as the solution, but it is very inefficient. Therefore, we employ a robust and efficient filtering step.

Filtering: Stationary points are those that do not comprise any movement on the user's part. Thresholding based on differential of the accelerometer data can be used to effectively remove these points.

To identify and eliminate stationary points in the accelerometer data, first step is to create a windows of size 5 around each candidate point. For each window, 20 DoGs (5 Octaves $\times$ 4 DoG) were created from the raw data by increasing the value of σ by $2^{\frac{1}{4}}$. All this together forms the 1st Octave. This is down-sampled by half to get the scale-space data for the next octave. We continue this process until we have data for 25 space-scales (5 Octaves $\times$ 5 space-scales). Now we compute the Difference of Gaussians for each consecutive space-scales to obtain 20 DoGs.

If the range of the value of DoG in each of these windows is less than a calculated threshold we discard this point as being a stationary point. For each of these DoGs computed, there exists a different optimal threshold value. An optimal threshold value should be such that any correct start and end points based on the ground truth are not discarded. Any threshold values that violates this is discarded.

Sliding Window: A heuristic based on the minimum and maximum possible duration of an eating cycle is then used to remove some more eating points. From experiments, we determined that the minimum windows size is about 1 second and the maximum is about 3 seconds and performed the sliding window. Also, all segmentation with a stationary point in between are discarded since the eating cycle is considered to be continuous. In a case that the user picks up food, but remains stationary for some time before eating, the algorithm simply picks the next point after the stationary point as the start time without compromising the total count.

2) Feature Selection: In preliminary experiments using laboratory settings (limited types of food, only eating related activities) we built a very accurate model for counting the number of eating cycles using a Support Vector Machine on eight statistical features of segmented windows. The eight features are min, max, mean, standard deviation, skewness, kurtosis, Root-Mean-Square, and Energy function. However when we used this model on a real-world meal data, the final results were not very good, possibly due to the presence

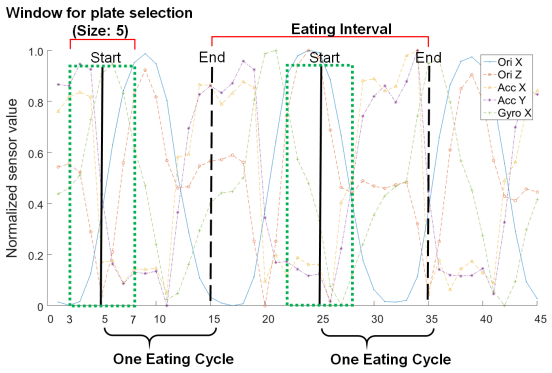

Figure 2: Eating activity segmentation.

of unrelated activities. As a solution, we developed a novel space-scale based feature selection. In this approach, there are four tasks: (1) DoG and Space-scale creation, (2) DTW score matrix generation (3) DoG and Space-scale selection, and (4) the statistical feature extraction.

DoG and Space-scale Creation: After segmentation, all segmented windows are compared to the ground truth. If the start time and end time of the window match exactly with that of the ground truth, the window is labeled as 'eating' otherwise it is labeled as 'other'. Then we obtain 25 space-scale time series and 20 DoG time series for a total of 45 different time series. For efficiency, we use only the first and last DoGs and space-scales data for each octave to obtain 10 DoGs and 10 space-scales. This is because for each octave the middle space-scales tend to be very similar. This process is repeated for all sensors resulting in 360 DoG and space-scales (18 sensors $\times$ (10 DoGs + 10 space-scales)).

DTW score matrix generation: To build this matrix, two different DTW comparisons are computed: (1) DTW between 'eating' and 'eating' window and (2) DTW between 'eating' and 'other' window. Then, we apply it to all 360 DoGs and space-scales. If there is 'N' number of 'eating' and 'M' number of 'others', we generate $C(N,2)$+(M$\times$N) rows $\times$ 361 (360 DoG and space-scale and label) columns matrix.

DoG and Space-scale Selection: Since linear features usually improve machine learning model performance, we select the DoG and space-scale that has the most linear characteristic between 'eating' and 'other' actions based on the DTW score matrix. We expect the scores between 'eating' and 'eating' windows to be lower than the score between 'eating' and 'other' windows. A DoG or space-scale feature which results in large difference in DTW score between 'eating' vs 'eating' and 'eating' vs 'other' actions is suitable for differentiating. To measure each DoG and space-scale linearity, a clustering evaluation method, Calinski-Harabasz index [9], is used. The scores between 'eating' and 'eating' are considered as first cluster and the scores between 'eating' and 'other' becomes other cluster. We perform this clustering evaluation to each column of DTW score matrix to evaluate the linearity of each DoG and space-scale. Based on the index value, we order the 360 different DoG and space-scale and used only top K. Depending on the value of K, the recall and precision vary. Through experiments we determined that

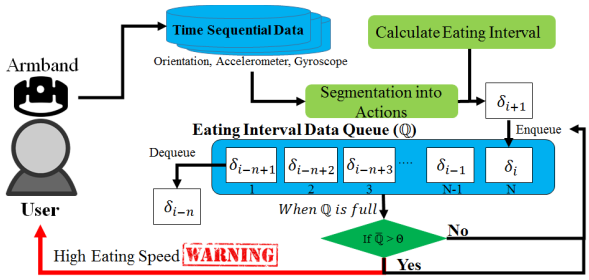

Figure 3: Queue based continuous eating speed feedback system. δ is one eating interval and θ is the interval threshold.

'$K = 80$' obtains the best precision and recall across all users.

Feature Extraction: After DoG and space-scale selection, we generate the eight statistical features of 3 segmented and whole time series from selected DoG and space-scale. Thus, the total number of final features becomes 2560 (80 DoGs and space-scales $\times$ 8 features $\times$ 4 segmentations).

B. Plate Section Identification

The input is data sets preprocessed from the sensor data that was originally collected from user's eating behavior. The output is to predict which section of plate that user picks. Since only start times of the eating cycles are useful features for this task, the dataset is preprocessed to obtain only those. The ground truth labels for each start time data were assigned by hand depending on the plate sections. Fig. 2 shows a window of size 5 around a start-point for the input.

C. Estimation of Food Portion

Food portion is estimated using EMG sensors of the Myo device. A given holding pattern results in activation of a specific subset of the eight EMG pods. For training we record the values for each of the 8 different EMG pods for a user when she is holding different weights. We tried 0g, 10g, 20g, 30g, and 40g weights and saved the data. From the previous steps we know the start time of the eating cycle. For each of the different weights, the EMG energy content is concatenated to form a single feature vector. For each individual, five databases are maintained corresponding to the 5 weight categories. Each database consists of $2^9 - 1$ potential EMG feature vectors. When a new data point arrives from the user, the 8 EMG pods and EMG energy is compared with every feature point in the database using DTW.

D. Physiological feedback based on eating speed

After segmentation into start and end times for an eating cycle, eating intervals are obtained by calculating the difference between end times of current and previous eating cycles as seen in Fig. 2. This can account for multiple pickings before eating, multiple eating from the same pick and other unrelated activities in between like moving of hands, talking or taking breaks. Identification of the eating interval allows us to provide eating speed feedback based as seen in Fig. 3. First, the eating interval (δ) is continuously stored in Q which is N size queue data structure until the user completes the meal. When the Q is full, the average of the interval data in Q called $\bar{Q}$ is computed. Then, it is compared with a predefined threshold time θ. If the $\bar{Q}$ is greater than θ, the

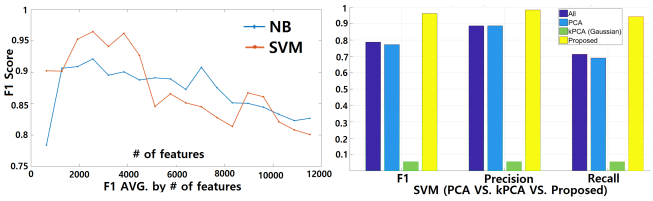

Figure 4: Left is SVM F1 score based on the number of feature. Right is comparison of dimension reduction methods.

user receives a warning related to her high eating speed and the next eating interval is stored in Q after the front item in Q is removed. If not, the next eating interval is stored in Q after the front item in Q is removed. This threshold is user dependent and chosen based on dietitian recommendations.

IV. Experimental Evaluation

Experiment Setup: We conducted two set of experiments; A) pilot experiment to study system functionality which was done in lab settings with limited types of food (e.g. bread and soup) on ten users, B) main experiment where we simulated real-life eating conditions using seven users. Subject is asked to sit facing a camera and commence to eat the food in their everyday manner. Subjects chose there preferred meal from Panda Express Chinese food chain. The plate had three different sections. Experiment starts with user performing the initial hand gesture. Once the gesture is made, the user has to hold her hand in such a way that the EMG pod interfaced with thermal and color camera faces the food plate. The user needs to hold the gesture for a maximum of 2.5 seconds. During this time, thermal and color images are taken from the plate and send to cloud for identifying food type and location. After identification is done, user can start the eating activity and the calorie count per eating action is reported in real time. We collect 18 data streams (sampled at 100 Hz) from the Myo wristband sensors. To build the ground truth each meal session is video recorded at 30 FPS.

Eating Cycle Identification: The method has three subtasks: 1) feature extraction methods, 2) experiment data types, and 3) supervised learning algorithms. For 1), there are two different feature selection methods: eight statistical features with four segmentations, all DoG and space-scale features, and our proposed feature selection. The number of eight statistical features with four segmentation and 18 sensors is 576, all 20 DoG and Space-scale is 11520 ($8 \times 4 \times 18 \times 20$), and our proposed feature selection method is 2560 ($8 \times 4 \times$ K (K=80)). For 2), as we mention in Sec. III, we used two different experiments: pilot and real-life. The performance of the pilot study was almost perfect when the SVM with all three different feature selection method. For the pilot study, precision was 1 and recall was 0.98. For real-life experiment, Tab. I displays the result. For 3), we performed three different algorithms: SVM, Naive Bayes, and Random Forest. In addition, we use DyFAV [10] which dynamically select feature and performs the linear classification to recognize American Sign Language using armband sensor [11]. Among these four classifications, SVM has the best F1. Also, Fig. 4 shows the our proposed feature selection method performance. When '$k = 80$', F1 with SVM is the best. Our method has much better performance compared to traditional feature dimension reduction methods such as PCA and kPCA.

Table I: Eating cycle identification performance.

Feature	Size	Algorithm	Mean Precision	Mean Recall	Mean F1
Eight Statics	576	NB	0.81	0.82	0.8
		SVM	0.87	0.71	0.75
		RF	0.77	0.55	0.62
		DyFAV	0.62	0.36	0.42
360 DoG & Space-scale	11520	NB	0.84	0.76	0.74
		SVM	0.9	0.73	0.8
		RF	0.57	0.42	0.47
		DyFAV	0.86	0.57	0.65
Proposed	2560	NB	0.86	0.77	0.78
		SVM	0.98	0.95	***0.96***
		RF	0.67	0.55	0.58
		DyFAV	0.91	0.63	0.7

Plate Sections & Food Portion: As mentioned in Sec. III-B, we used five plate sections. We performed NB, SVM, and MLP for identification the plate sections that user picked. The average accuracy is the 99% when we use MLP. To estimate the food portion, we measured the food weight using EMG sensor based DTW. The average accuracy is 90.49%.

V. Conclusion

FIT-ADAM&EVE is an automated diet monitoring system that uses an armband to monitor food type, and eating speed in terms of calories per bite with high precision and recall. It can also determine food portion sizes with an accuracy of almost 90% through the use of EMG sensors. It is triggered by a single customizable gesture performed by the user, which is the only input required for eating activity monitoring. It is real time and can provide consumed calorie count after the meal. *FIT-EVE&ADAM* is a user-friendly diet monitoring system that is expected to promote healthy eating habits.

References

[1] Raymond C Baker and Daniel S Kirschenbaum. Weight control during the holidays: highly consistent self-monitoring as a potentially useful coping mechanism. *Health Psychology*, 17(4):367, 1998.

[2] Parisa Pouladzadeh, Shervin Shirmohammadi, and Rana Al-Maghrabi. Measuring calorie and nutrition from food image. *Instrumentation and Measurement, IEEE Transactions on*, 63(8):1947–1956, 2014.

[3] Pallavi Kuhad, Abdulsalam Yassine, and Shervin Shimohammadi. Using distance estimation and deep learning to simplify calibration in food calorie measurement. In *Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2015 IEEE International Conference on*, pages 1–6.

[4] Mingui Sun and et al. ebutton: a wearable computer for health monitoring and personal assistance. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6. ACM, 2014.

[5] Junghyo Lee, Ayan Banerjee, and Sandeep K. S Gupta. Mt-diet: Automated smartphone based diet assessment with infrared images. In *PerCom*, pages 1–6. IEEE, 2016.

[6] Austin Meyers and et al. Im2calories: towards an automated mobile vision food diary. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1233–1241, 2015.

[7] Joohee Kim and et al. Slowee: A smart eating-speed guide system with light and vibration feedback. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 2563–2569. ACM, 2016.

[8] Sougata Sen and et al. The case for smartwatch-based diet monitoring. In *Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 585–590. IEEE, 2015.

[9] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[10] Prajwal Paudyal, Junghyo Lee, Ayan Banerjee, and Sandeep KS Gupta. Dyfav: Dynamic feature selection and voting for real-time recognition of fingerspelled alphabet using wearables. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 457–467. ACM, 2017.

[11] Prajwal Paudyal, Ayan Banerjee, and Sandeep KS Gupta. Sceptre: a pervasive, non-invasive, and programmable gesture recognition technology. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 282–293. ACM, 2016.