10-2020

Annapurna: An automated smartwatch-based eating detection and food journaling system

Sougata SEN
Singapore Management University, sougata.sen.2012@phdis.smu.edu.sg

Vigneshwaran SUBBARAJU
Singapore Management University, vigneshwaran@smu.edu.sg

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

Rajesh Krishna BALAN
Singapore Management University, rajesh@smu.edu.sg

Youngki LEE
Singapore Management University, YOUNGKILEE@smu.edu.sg

Citation

SEN, Sougata; SUBBARAJU, Vigneshwaran; MISRA, Archan; BALAN, Rajesh Krishna; and LEE, Youngki. Annapurna: An automated smartwatch-based eating detection and food journaling system. (2020). *Pervasive and Mobile Computing*. 68, 1-19.
Available at: https://ink.library.smu.edu.sg/sis_research/7155

Annapurna: An automated smartwatch-based eating detection and food journaling system

Sougata Sen[a*] Vigneshwaran Subbaraju[b] Archan Misra[c] Rajesh Balan[c] Youngki Lee[d]

[a] Northwestern University, 2145 Sheridan Ave, Evanston, IL 60208, USA * corresponding author

[b] A*STAR, 1 Fusionopolis Way, 138632, Singapore

[c] Singapore Management University, 80 Stamford Road, 178902, Singapore

[d] Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, South Korea

Published in Pervasive and Mobile Computing, 2020, 68, 101259, pp. 1-19.
DOI: 10.1016/j.pmcj.2020.101259

Abstract

Maintaining a food journal can allow an individual to monitor eating habits, including unhealthy eating sessions, food items causing severe reactions, or portion size related information. However, manually maintaining a food journal can be burdensome. In this paper, we explore the vision of a pervasive, automated, completely unobtrusive, food journaling system using a commodity smartwatch. We present a prototype system — Annapurna— which is composed of three key components: (a) a smartwatch-based gesture recognizer that can robustly identify eating-specific gestures occurring anywhere, (b) a smartwatch-based image captor that obtains a small set of relevant images (containing views of the food being consumed) with a low energy overhead, and (c) a server-based image filtering engine that removes irrelevant uploaded images. Through lessons learnt from multiple user studies, we refine Annapurna progressively and show that our vision is indeed achievable: Annapurna can identify eating episodes and capture food images (involving a very wide diversity in food content, eating styles and environments) in over 95% of all free-living eating episodes.

Keywords

Wearable sensing, Mobile computing, Food journaling, Automated eating tracking system, IMU and camera data processing

1. Introduction

Automating the creation of a personal food diary has been a long-standing research goal in the mobile sensing community for supporting various wellness-related goals – e.g., losing or maintaining target weight, or capturing unhealthy eating habits, such as eating fast or midnight snacking. However, most proposed automated solutions are either obtrusive – e.g., hanging the phone around the neck [1], clipping a camera to the clothing [2], or rely on specialized wearable sensors – e.g., ear-worn wearables [3], [4], [5], or neck-mounted devices [6], [7], [8]. More recently, researchers have utilized the inertial sensors on a more mainstream wearable (a wrist-worn smartwatch) to capture the 'intake' gesture during eating [9], [10]. Such wrist-worn approaches hold promise as an unobtrusive, automated way to identify eating episodes, but they do not capture images of the food being consumed — i.e., do not help answer the question "What did you eat?".

To bridge this gap, in this work we demonstrate the possibility of combining inertial sensing signal (to detect specific eating gestures) from a smartwatch with a smartwatch-mounted camera to capture & curate useful images of the food

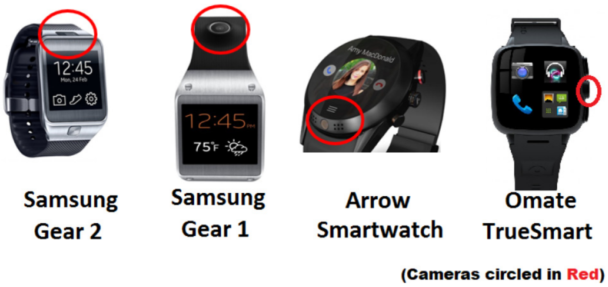


Fig. 1. Smartwatches with embedded cameras located at various positions.

items consumed, without imposing a prohibitively high energy overhead. Our work is motivated by the gradual emergence of smartwatches with embedded cameras, some of which are shown in Fig. 1, as well as the gradual adoption of small form-factor, wearable camera devices (such as Narrative Clip [11] & SenseCam [2]), which can record image and sensor data for an entire day while running on batteries. In addition, advances in deep learning-based food recognition (e.g., [12,13]) have increased the intrinsic value of such captured food images, to support applications such as dietary conformance analysis and personalized food recommendations. Our core idea is simple: (a) a smartwatch's inertial sensor can help identify the eating-related "hand-to-mouth" gestures; while (b) the embedded camera can then opportunistically take appropriate pictures, when it has a clear, unobstructed view of the food being consumed. We investigate these possibilities and show that such unobtrusive, gesture-triggered capture of food content for a journal is not just feasible, but also appeals to real-world users. We develop the necessary technical components and embed them to realize a prototype system – *Annapurna*,[1] which automatically provides an individual with a portal-based view of the food items consumed during the day (as well as other gesture-related information such as number of mouthfuls eaten).

In this work, we focus primarily on automatically capturing images of food consumed during meals and served in plates. Indeed, as we shall see in Section 3, 80% of meals consumed in Singapore involve plated food items, which are amenable to capture by *Annapurna*. We suspect that this predominance of plated food will be true in eating cultures across several parts of Asia where rice is the core item of the meal component [14]. Of course, in cultures were plated meals are less common (e.g., cultures where food items such as sandwiches that can be consumed without a plate are observed frequently), *Annapurna* can still recognize the onset of an eating episode, but will not be able to capture images of the consumed food. We discuss about an alternate journaling strategy in such settings in Section 10.

Overall, our vision gives rise to these **key research questions:**

- *Can the inertial sensors on the smartwatch be used to reliably identify eating gestures early enough to trigger the camera?* Continuous image capturing using a camera is energy intensive [15]. Thus, it is necessary to develop techniques that opportunistically trigger the camera. Recent works, including ours have shown that individual eating gestures may be identified using the smartwatch's inertial sensor data [9,16]. However, in these approaches, the eating gestures are detected only after the gesture is over, making them unsuited to the goal of triggering the camera to capture the images. To obtain useful images of the food, individual eating gestures must be identified as soon as the gesture starts. An open question thus relates to the robustness vs. latency characteristics of eating gesture recognition — i.e., can eating activities be identified reliably and fast enough to even capture relatively short-lived eating episodes? This is a formidable challenge as real-world eating is inherently a very diverse activity, distinguished by (a) mode of eating (with hands, chopsticks, forks etc.), (b) food type (e.g., noodles, pizza, sandwiches, etc.) and (c) venue-specificity (e.g., relative height of table vs. chair, shape of plate).
- *Can the smartwatch's camera acquire useful (good-quality) images of the food being consumed?* Since our goal is to unobtrusively capture images of food that an individual consumes, the smartwatch camera must acquire these images during the eating gesture. This is a significant challenge since the user's wrist is in motion, leading to blurry images, images of other-person's food, etc. Further, commercial smartwatches have their camera in different positions, some of which are unsuitable for capturing images of the food.
- *Can the sensing on the wearable be suitably optimized?* While continuous background monitoring of inertial sensors has now become feasible [17], continuous use of the camera sensor (to capture possible food images) is energy intensive [15]. To support continuous day-long operation, we must selectively trigger the inertial sensors and the camera. This appropriate time window must be *learnt*, so that it can accommodate diversity in eating styles.
- *Can we automatically curate the captured images?* Even if the smartwatch's camera is triggered only during eating episodes, the captured set of images may contain a significant fraction of *irrelevant* images — i.e., non-food related images or blurry images. It is important to automatically eliminate such images, both to save bandwidth/energy on the wearable and mobile device, and to reduce the user's cognitive load. Moreover, such curation needs to take into account

[1] Video showing the system's working is available at http://is.gd/annapurna.

real-world characteristics of eating gestures, such as potential long pauses between successive hand-to-mouth gestures. In addition, the execution of such a curation pipeline must be carefully partitioned across multiple platforms (the smartwatch, the usually-paired smartphone and the back-end cloud infrastructure) to best balance energy-vs-accuracy tradeoffs.

To address these research questions, we designed and evaluated *Annapurna*, a system that unobtrusively and automatically logs and filters images of food consumed in a plate during a meal. *Annapurna* has a smart-watch component which automatically identifies eating episodes, trigger the camera at the right time to capture images of the food plate in a energy efficient manner. *Annapurna's* backend system identifies relevant images and logs them as a food journal for the end user. Our design is driven by extensive real-world observations of eating behavior and validated through a series of studies, conducted within the city of Singapore: one in-lab controlled study, one directed semi-controlled free-living study, and two completely free-living studies. Every study helped in improving subsequent studies, thus improving the robustness of *Annapurna*.
Key Contributions: The key contributions in this paper are:

- *Robust, real time identification of eating gestures*: Using detailed studies on real-world eating gestures recorded from subjects in Singapore, we designed and developed a 2-tier robust classifier for recognizing eating episodes (details in Section 5). This classifier balances the requirement of not missing eating episodes while reducing the energy overhead (recall vs. energy tradeoff) by: (a) requiring multiple atomic eating gestures (classified over 500 ms segments) to determine the onset of an eating episode (as a precursor to activating image acquisition by the camera) and (b) using the absence of subsequent gesture segments to aggressively deactivate the image acquisition process. This step allowed recalling 95% eating episodes in the two free-living studies.
- *Feasibility of capturing useful images of the food consumed*: In Section 3, we demonstrate that we can automatically capture useful food images using a commercial smartwatch's camera in 65 to 90% episodes. However, this success depends on the position of camera on the watch. Additionally, we demonstrate that the right moment (during an eating gesture) for capturing such an image is position-dependent. We also demonstrate how a pipeline of computationally-simple image processing techniques can eliminate irrelevant images.
- *An optimized image capturing process, balancing battery-load while ensuring that useful images are captured*: In Section 6, we show how the *Preview* mode achieves the best balance between the goals of minimizing energy-intensive [15] smartwatch-based image acquisition and capturing usable food images: this mode takes a sequence of pictures with low energy overhead (40.7 mW per image) and can be triggered with low (less than 435 ms) latency. We also evaluate two different strategies to stop the image acquisition pipeline by exploiting the trade-off between energy and efficacy. In Section, 5.4, we further show that simple heuristics may be used to further reduce the battery load of the overall system.
- *Evaluation of Annapurna in free-living conditions*: Through multiple in-the-wild studies collected from 8 individuals, each contributing at least 4 days of eating data (described in Section 9), we demonstrate that *Annapurna* could correctly capture the in-the-wild eating episodes and display useful food images in the portal with a precision and recall of 95% each. These studies are substantially longer than several existing in-the-wild, unconstrained eating detection studies. However, further evaluation with more longitudinal studies on demographically diverse population would be necessary to potentially fine tune the system to region or culture-specific food choices and eating habits.

2. Related work

Food journals such as MyFitnessPal allow users to manually record all food items that they consume through the day [18]. However, self reporting is onerous as well as vulnerable to reporting bias [19,20]. To overcome the challenges associated with manual food journaling, several researchers have proposed automated food consumption monitoring systems, and journaling approaches. Automated journaling uses techniques such as instrumented locations [21], modified tabletops [22], or even utilizes off-the-shelf [16] or custom made [23] mobile and wearable devices. Indeed, researchers have explored numerous types of wearables such as wrist-worn [9,24], ear-worn [4,5], neck-worn [7,25], or eye-worn [26,27] devices for eating detection. For each type, researchers have explored various sensor signals such as utilizing sensor data from inertial [9,23], audio [4,7], image & video [1,28], or even a fusion of multiple sensor data [3,29]. In addition to automatically detecting eating activity using images and videos, researchers have used images to identify the food items that an individual consumed [30,31]. Since we rely on a smartwatch for detecting the eating activity, in this section, we introduce relevant eating detection and monitoring systems which primarily utilize similar devices. We specifically discuss about (a) inertial sensor for eating gesture identification, (b) automatically capturing images of food, and (c) analysis of the food images.

2.1. Inertial sensor-based eating detection

Amft et al. demonstrated the possibility of detecting eating gestures by evaluating signals from four on-arm accelerometers [32], while Dong et al. used the gyroscope of a custom-made wrist-worn device to detect similar gestures [23]. With the advent of commercial wearable devices, especially smartwatches, with embedded inertial sensors, they find natural utility in detecting the eating activity. Ye et al. conducted a controlled study and demonstrated the possibility of utilizing accelerometer data from a smartwatch and a smartglass to determine eating gesture [33]. The fusion of both the accelerometer and gyroscope signal from a custom device for eating gesture detection was also explored by Dong et al. [34]. Clearly, these studies demonstrated that inertial sensors could facilitate monitoring the eating activity in controlled settings. To explore the possibility of monitoring the eating activity in free-living setting, Thomaz et al. conducted a user study where they analyzed sensor data from a smartwatch's accelerometer in free-living conditions [9].

Although the previously mentioned studies relied on a single wearable device, a multi-device approach has the potential to capture finer details of the eating activity. Mirtchouk et al. explored this possibility of using multiple sensing devices and observed that the performance indeed improved [29]. A major challenge with these studies is that the systems detected the individual eating gesture (or episode) post hoc, i.e., after the gesture (or episode) completes. This makes them unsuitable for capturing food images by opportunistically triggering the camera at the right moment within a eating gesture, during a eating period. In our work we try to tackle this additional challenge by building upon these earlier works and also investigate the interplay between the dynamics of such inertial sensing and the real-time camera based capture of relevant food item images.

2.2. Capturing images of food consumed

Although several researchers use image and video data for ground truth and validation purposes [4,25], however, an alternate approach for food intake identification and monitoring is by utilizing and processing these visual information, i.e., the images or videos. The first step towards such monitoring is to successfully capture images of the food item that an individual is consuming. Several researchers demonstrated the feasibility of capturing food image using a smartphone's camera [1,35]. The phone was suspended across the user's neck using a lanyard, however makes the system obtrusive. Zhu et al. removed the obtrusiveness by asking the users to manually capture the image of the food plate at the start and end of a meal [36]. However, this approach trades-off the automaticity of the system for unobtrusiveness. Recognizing the food consumed by the user from the images obtained from the camera of a smartphone has been studied by Kawano et al. [37] and Lee et al. [30]. However all these techniques require the user to explicitly acquire or label the images of the food and then they identify the food item.

2.3. Automated analysis of images

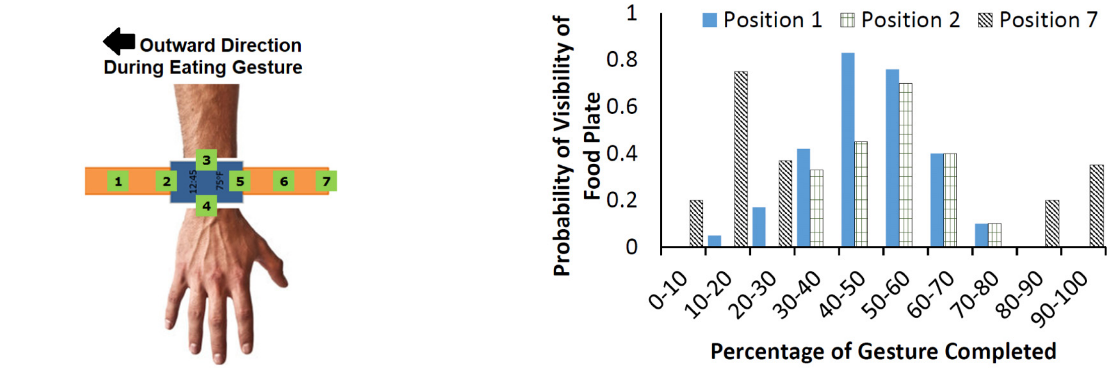
Recently, researchers have explored analysis of images to automatically detect the eating activity [38,39]. However, the images in these studies have been captured in controlled laboratory environments. A major challenge in free living studies is to automatically identify and filter out images that do not contain food items, both to improve the analysis accuracy and also to preserve the energy consumption of wearable devices. Several prior works have also discussed the possibility of filtering the relevant food-related images (from the large set of images acquired from wearables) using standard image processing methods [1,3,40]. More recently, with the popularity of deep learning techniques for image processing, several researchers have proposed techniques to automatically identify images of food items [13,41,42]. The images acquired in these studies are focused on ego-centric first person views which are considerably less challenging as compared to the images acquired from a smartwatch during an eating gesture. However, such techniques can utilize our work which effectively performs pixel isolation to extract such frontal views from images captured automatically during eating gestures.

3. Feasibility of food journaling using a smartwatch

We envision *Annapurna* as an unobtrusive diet journal. To this end, we investigate (a) what relevant aspects of real-world eating activities do we need to incorporate in the design of robust classifiers for eating detection? and (b) can a smartwatch camera even capture images of the food being consumed? We first investigate whether this depends on the type of food or the on-watch placement of the camera sensor via a comprehensive controlled microscopic study.

3.1. Controlled-study details and insights gained

We performed an extensive *micro-study* with 21 participants (8 females, 13 males), employed in our university. These participants were involved in a total of 135 eating episodes that were consumed in public food courts. We define an eating episode as *the period of time between the start of a meal (i.e., the meal is either placed in front of an individual in a restaurant or handed to the individual in a self-service eatery) and the consumption of the last spoonful*. The participants consumed these meals during regular meal hours when they went for lunch, snacks or dinner. Most episodes took place in the university's food court (self-serviced and illuminated by artificial lights), with a few occurring outdoors. A custom application running on the watch (Samsung Gear 1 smartwatch worn on the eating hand) collected accelerometer, gyroscope and preview frames during the entire episode, while an external observer video-recorded the meal (for visual confirmation). Separately, we also recruited 2 individuals to investigate the sensitivity of the results to the on-body location & orientation of the smartwatch camera. The users wore three distinct smartwatches, Samsung Gear 1, Samsung Gear 2 and Omate TrueSmart



(a) Different positions of camera.

(b) When is the food plate visible?

Fig. 2. The possibility of capturing images of the food plate from various smartwatch-embedded camera position.

Table 1
Key results from the controlled in-lab study.

Food item	Eating Modality	# of episodes	Completion time (s)			Hand to mouth Gestures (count)			Episodes with useful
			min	max	avg	min	max	avg	Frames
Rice	F,S	66	211	1140	568	22	54	33.5	95.5%
Sandwich	H	20	255	363	299	6	35	14.4	65%
Pasta/noodles	F/C	29	234	771	459	13	35	27.3	86.2%
Fruits	T	20	51	387	183	7	23	13.5	70%

F: Fork, *S*: Spoon, *H*: Hand, *C*: Chopstick, *T*: Fruit pick.

(illustrated in Fig. 1), each with the camera mounted in a distinct position on the outward or inward rim of the watch bezel or on the strap. By varying the orientation on the wrist, we obtained 7 different camera positions (Samsung Gear 1 for positions 1,6 and 7; Samsung Gear 2 for positions 2 and 5; and Omate TrueSmart for position 3 and 4), as illustrated in Fig. 2(a).

Key Insights: Table 1 highlights some of the key parameters associated with the consumption of these food types. Some key observations were:

Duration of the meal: We observed wide variations in eating gestures for the different food types considered. For items like rice and noodles, episodes lasted anywhere between 3.5 to 19 min, involving 13 to 54 separate hand-to-mouth gestures.[2] Among these food items, we also observed that: (a) sandwiches and fruits presented the least number of distinct hand-to-mouth gestures (as users often held the items close to their mouth between successive bites), (b) noodle or pasta had high variability in the number of hand-to-mouth gestures mainly due to the use of forks vs. chopsticks, while (c) the variation for rice-based meals is generally due to the individual eating speed and quantity consumed in each mouthful.

Possibility of image capture & orientation sensitivity: On observing the annotated images captured by the smartwatch camera we found that the likelihood of obtaining *at least one usable food image* (one which provides an unobstructed view of the consumed food) is fairly high (80% or higher) for all food types, except for sandwiches and fruits, i.e., in situations where the user never puts the item down on the plate.

We then analyzed the videos captured from the 2 users who wore the different smartwatches which had the cameras in different positions. The food plate was visible at least once (for both users) only for camera positions 1,2 and 7. More specifically, a useful image is found in 82.6%, 77.4% and 80.4% of all eating gestures, respectively, for these positions. Moreover, Fig. 2(b) shows the probability of the images being *useful* (i.e., the food item is visible) as a function of different points in the gestural sequence (the 50% point corresponds roughly to the zenith, where the hand is closest to the mouth). We see that the on-watch camera position significantly affects this probability — for *position 1 & 2*, the plate is most visible when the hand was near the mouth (which, incidentally, is also the point where the hand movement speed is the lowest). To provide a clearer understanding of the images captured at different points in the gesture, in Fig. 3 we present images captured from an entire representative eating gesture when the camera was positioned at position 1. Our studies help establish guidelines on where the image sensor must be placed on wrist-worn wearables, to support such diet capture.

Additional Background Data & Implications: Clinical studies on women with normal BMI showed that the mean duration of breakfast, lunch, and dinner is 14, 28 and 29 min respectively [43]. Also, survey-based studies on 80 secondary

[2] *Annapurna's* energy-efficient food image capture technique leverages on the observed presence of tens of such gestures in an eating episode.

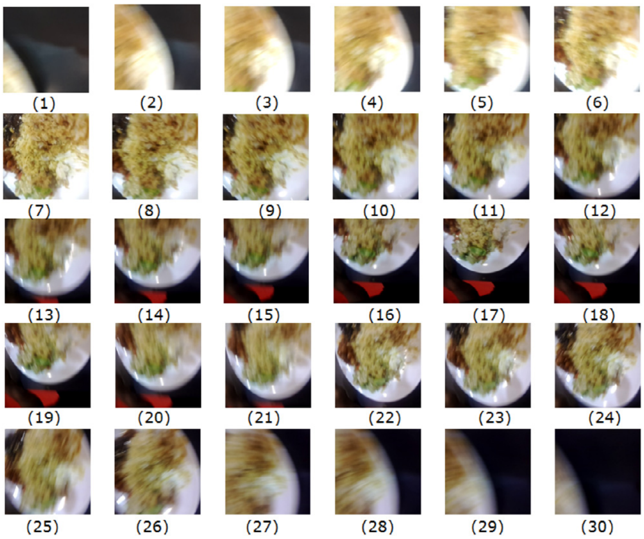


Fig. 3. Sub-sampled image frames extracted from the video captured when the camera was at position 1. The entire gesture duration was 4 s. In this gesture, the useful images were obtained from the 20% to 40% of the gesture duration.

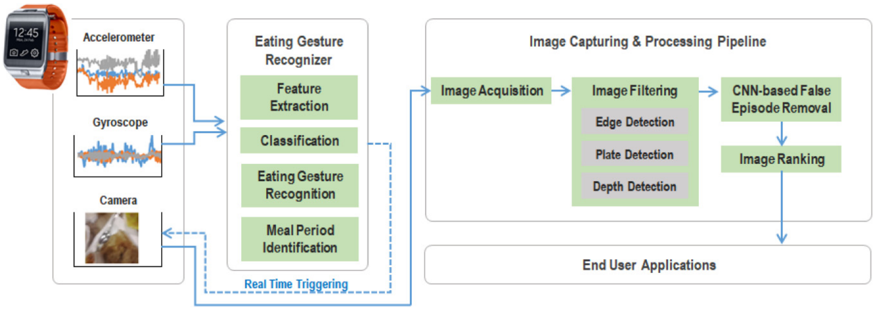


Fig. 4. *Annapurna* consists of an Eating Gesture Recognition module that utilizes inertial sensor data to detetct eating gestures. On detecting eating gestures, it triggers the camera to capture images of the food plate. The captured images are passed through an image processing pipeline to detect images of teh food plate. These images are then presented to the user in the form of a food journal .

school boys in Singapore revealed that over 80% of the standard meals (lunch and dinner) consisted of plate-based items (e.g., noodles, rice) with only 8% or less being 'one-item' meals (e.g., chicken nuggets, burgers and sandwiches) [44]. Hence, we believe that in the Singaporean (and perhaps, more broadly, in the Asian) context, it is meaningful to develop a system that (a) principally focuses on plate-based food items, and (b) assumes that an eating episode will contain dozens of hand-to-mouth gestures. Indeed, *Annapurna* is designed to capture the food consumed during 'significant eating episodes' (meals, which involve multiple hand-to-mouth gestures), and does not target the isolated acts of eating (e.g., popping a piece of candy into one's mouth).

4. System architecture

Fig. 4 provides the high-level workflow of *Annapurna*. Broadly, the components on the smartwatch detect the intermittent *eating activities* during the day, and selectively triggers the camera to capture *likely* images of the food consumed. Subsequently, these images are subjected to *filtering* on the user's paired smartphone to remove some irrelevant images. The remaining images are subjected to comprehensive processing on the server to *rank* and select a small but precise set that best represents the food associated with an eating episode. Finally, these images, and other relevant eating-related information, are displayed to the user. *Annapurna* comprises of the following components:

1. *Detecting the eating gesture:* The *Eating Gesture Recognizer* module on the smartwatch uses the accelerometer and gyroscope sensor data to detect (i) the onset of an eating episode, and (ii) individual repetitive eating (i.e., hand-to-mouth) gestures within the episode. It balances recall (missing eating episode) and precision (classifying non-eating activities as eating). Additionally, it should identify eating episodes fairly early (i.e., within a few hand-to-mouth gestures), to capture short-lived meals. We describe the implementation details of this module in Section 5.

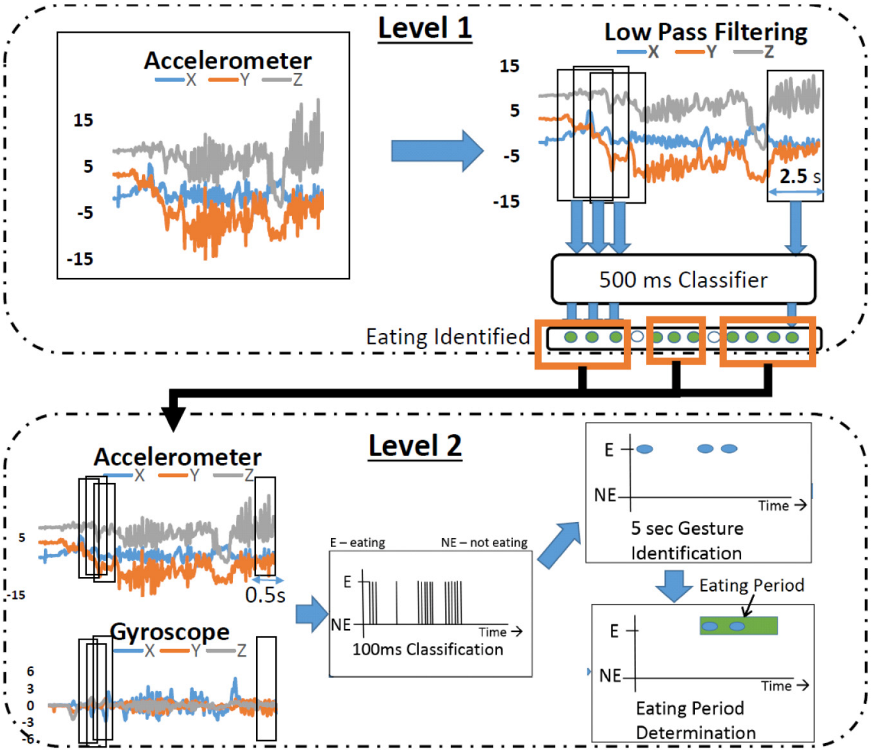


Fig. 5. Pipeline for recognizing the eating gestures and episodes. Level 1 uses a longer 2500 ms window with 80% overlap; Level 2 uses a shorter 500 ms window with 80% overlap.

2. *Responsive image capturing:* On identifying the onset of an eating episode (via multiple closely-spaced eating gestures), *Annapurna* captures images automatically by triggering the camera at the right time during subsequent gestures. This is challenging because eating gesture is relatively short-lived, and the smartwatch's latency for image acquisition (i.e., time to turn on and capture an image) is relatively high (≈0.9 s). In Section 6, we show how a preview-mode based image capture strategy overcomes this challenge.
3. *Image filtering:* This component performs (a) irrelevant image elimination, and (b) selection of the *best* set of images for each eating episode. For energy-efficient operation, the smartphone executes only simple (but effective) image pre-processing that eliminates a bulk of the images; the backend server ensures that relevant images are identified and displayed to the user. In Section 7, we detail these image elimination and ranking techniques.
4. *Food journaling:* Finally, the server stores this small subset of *relevant* images corresponding to each detected eating episode. Users can view these images via a Web portal. While the portal development is straightforward, in Section 8, we discuss some design choices (e.g., number of images per episode to be presented) intended to improve the overall *user experience*.

5. Detecting eating gestures

Our final design of the two-level eating detection pipeline is shown in Fig. 5. We first describe the initial implementation of this pipeline, and then describe its evolution based on experiences gathered from real-world studies.

5.1. Feature extraction and classification

We extracted the raw accelerometer and gyroscope data and manually labeled the hand-to-mouth gesture periods in the data. We observed that an average eating episode has 18 to 19 eating gestures. Our initial approach was to use features defined over short frames of 500 ms for both accelerometer and gyroscope data. The small frame size is needed to trigger the camera reasonably quickly to get appropriate images. This approach is shown in the bottom part (Level 2) of Fig. 5. The raw sensor data is partitioned into frames of length 500 ms (with 80% overlap between frames); a set of widely-used time and frequency domain features for the three axes of both accelerometer and gyroscope (identical to features extracted in [45]) are then derived for each frame. We built a *person-dependent* classification models. Table 2 shows the accuracy, precision and recall of a 10-fold cross-validation for three commonly used classifiers. From the table we observe that both

Table 2
Performance of classifiers in identifying eating gestures.

Classifier	Accuracy	Precision	Recall
DT	96.6%	96.1%	96.5%
RF	98.2%	97.1%	99.0%
SVM	85.7%	83.6%	87.1%

Table 3
Gesture prediction error (%) for different window size(w), threshold (t).

w	t (count)				
(s)	10	20	30	40	50
2	−152.1	–	–	–	–
5	−4.2	−22.2	−3.4	–	–
10	48.3	35.7	34.3	35.9	33.9

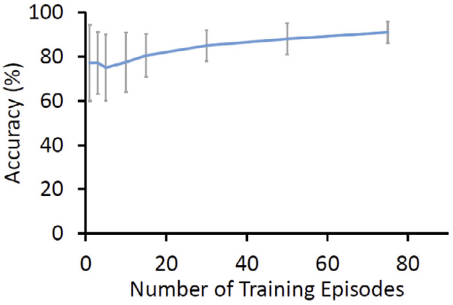


Fig. 6. Variation of accuracy as training data size is varied.

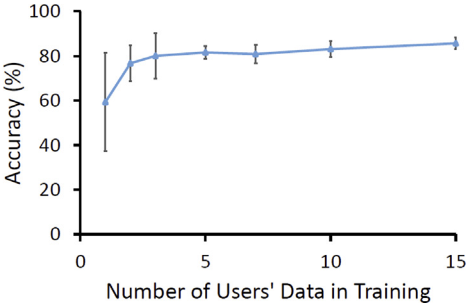


Fig. 7. Variation of accuracy as number of users is varied.

Decision Tree and Random Forest classifiers offer high classification accuracy. We eventually selected the Decision Tree classifier because its lower computational complexity makes it more suitable for a resource constrained smartwatch.

To understand whether the performance of the decision tree-based eating detection model is influenced by the size of the training data, we considered the 95 episodes where participants either consumed rice or noodles/pasta (66 rice, 29 pasta/noodles). We divided these 95 episodes into training and test set; we explored various training-test set size ratios. Episodes were randomly assigned to either the training set or the test set, while ensuring that the training-test ratio was maintained. The training set was used to create the decision tree-based eating gesture detector and this model was tested on the test set. We repeated this process of randomly assigning episodes to training or test set 10 times. Fig. 6 shows the average performance of the classifier for various training set sizes. From the figure we can see that for n = 1, the prediction accuracy is 77%, which appears to be reasonably high. However, on scrutinizing the prediction results, we observed that for 36% of the experiments, every instance in the episode was predicted as not-eating. This percentage of episodes where all instances are predicted as not-eating drops to 10% when we use 15 episodes for training and further to below 5% when 30 episodes are used for training. At the same time, the prediction accuracy when using 30 episodes for training is above 85%.

We next analyzed the performance of a person-independent model when all episodes of participant $p : p \in \{1, 2, 3, 5, 7, 10, 15\}$ was used to create the training model and it was tested on the remaining participants. This analysis was performed on the 95 rice and noodle/pasta consumption episodes. For the training data, we randomly selected p

Table 4
Variation of false positives and false negatives for different values of cost parameter used in building a cost-sensitive decision tree for detecting eating activity.

	0	20	35	50	100
False Positives	36.6	18.9	12.6	8.6	6.7
False Negatives	3.5	8.9	17.4	37.1	55.3

participants from the pool of 21 participants. For every value of p, we repeated the process 10 times. Fig. 7 shows the variation in performance when the value of p is varied. From the figure, we also see that there is significant variation in performance at lower values of p. This indicates that the performance of the system is affected by the users selected for building the model. However, for $p \geq 5$, the variation is low. Additionally, beyond $p = 5$, the performance of the system stabilizes, thus indicating that building a robust person-independent model is indeed possible.

For a 500 ms window of sensor data, we observed that even during an eating gesture, two consecutive frames were not always classified as *eating*. There were also periods during the eating episode when *non-eating* gestures (e.g., adjusting one's hair or waving) were classified as eating in several 500 ms windows. On average, our classifier's prediction indicated that during a single eating episode, there were 337 transitions from *non-eating* to *eating*. This is much higher than the ground-truth (average of 18–19 gestures), indicating the need of a second window to remove the noise.

5.2. *Determining length of an eating gesture*

From the ground truth data we observed that an average eating gesture lasted for 3.1 s (Rice - 2.8 s, Noodles - 3.7 s, Sandwich 3.1 s) where a gesture starts from the point the hand starts moving upwards and ends when the hand comes back to rest. To evaluate the 500 ms window's performance in detecting eating, we take a window (w) of past raw classifier outputs (obtained every 100 ms.) and compare the number of *eating* gestures identified by the classifier during this window with a threshold (t) value. If the number of positive classifications in w is more than t, then we declare the window to be an eating gesture window. Table 3 presents the average error in determining the number of gestures (transitions from *not-eating* to *eating*) in an episode, as a function of w and t. We computed Prediction Accuracy=$\frac{\sum GT - \sum P}{\sum GT} * 100$, where GT is the number of actual eating gestures and P is the system-predicted gesture count. A positive value indicates that our system is under estimating, while a negative value indicates over-estimation. From this table, we see the lowest values of error in gesture estimation are obtained for $w = 5$. A smaller value ($w = 2$ s.) over-estimates the number of eating gestures, whereas an overly large window ($w = 10$ s.) under-counts the eating gestures.

When we compared the estimation errors for different settings of w and t for individual food items (rice and noodles), we found that they are indeed different, due to the different eating styles. (In case of noodles, the user usually holds the hand near the mouth till she has consumed the entire strand of noodle.) However, even though t and w varied across different food items, the variation was modest enough to allow us to use $t = 10$ and $w = 5$ across food-types.

5.3. *Refining the classifier*

We observed that during a rice eating episode, an eating gesture occurred every ≈17 s. However, the gesture occurrence was not evenly distributed, but was rather bursty. Since we wanted to capture images of the food plate when we determined eating, we aimed at capturing the image early in the episode so that the food items were still on the plate. On average, the first minute of the rice eating episode had ≈ 3 eating gestures. To ensure high recall, we chose a conservative approach and declared the start of an eating episode if at least 2 gestures were detected in a minute.

Step 1 — Building a Cost-Sensitive Classifier: When the base classifier (described above) was applied to the controlled study dataset, it resulted in a high positive rate. This triggered detection of many false eating episodes and drained the battery rapidly by turning on the camera unnecessarily. To tackle this problem, we increased the cost of false-positive mis-classification in the training phase, thereby building a cost-sensitive classifier. However, this new classifier missing several real eating episodes in a feasibility study (low recall).

Step 2–Cost-Sensitive, Two-stage Classifier: To address this challenges, we needed: (a) to determine the optimum cost for the classifier to trade-off between false positives and false negatives, and (b) an additional pre-classifier, that worked on a longer window, to reduce the false-positives.

We performed a grid search and evaluated the cost-sensitive decision trees to determine the optimum cost parameter. Additionally, we acquired day-long regular life-style sensor traces of non-eating activities from 3 participants who were asked to remove their watches when they were eating, but wear the watch at other times. For the models with different cost parameter settings, the false-negative rate was determined from cross-validation on the controlled-study training dataset itself. To evaluate the false-positive rate, we used the day long traces of non-eating data (from these 3 participants). Table 4 provides the false-positive and false-negative rates for different values of cost parameter. When the cost of false detection is low, the *FN* rate is low, meaning we will not miss many eating gestures. However, the *FP* rate on real-life

trace is very high (36.8%). For a cost of 100, the *FP* rate on the real-life trace is very low (6.7%), but the *FN* rate for eating is also very high (55.25%), implying we will miss a substantial number of eating episodes.
From this table, we observe that a cost parameter of 35 provides a suitable value for both *FP* rate (12.6%) on the real-life trace and the *FN* rate (17.4%) for detecting eating gestures.

We also observed that several false-positives were generated by "jerky movements" of the hand during regular activities such as gesticulating during interactions or repeated lifting of objects. While a small frame-duration of 500 ms is needed for efficient, low-latency triggering of the camera, an additional longer-frame duration of 2.5 s was also needed to eliminate these transient gestures. Accordingly, we developed an additional classifier (Level 1, as shown in Fig. 5) that uses a longer 2.5 s. frame of accelerometer data alone, to first identify the *likely* eating episodes. As each eating episode is long-lived, this initial classifier triggers the fine-grained classifier (Level 2 in Fig. 5) which works on the shorter 500 ms frames, additionally using the gyroscope readings also. Once the eating gesture is consistently detected in level 1 (for more than 10 frames within a minute), this triggers the cost-based classifier (described earlier) that operates on 500 ms frames.

5.4. Additional heuristics to reduce energy consumption

Based on our observation of data, we identified several situations where we could turn off all or some of the sensors, allowing reducing energy consumption, resulting in longer battery life. Additionally, from prior literature, we identified several Singapore-based lifestyle characteristics that could further reduce the energy consumption. We next introduce these energy saving heuristics that we applied in *Annapurna*.

- If *Annapurna* detects that the hand has been at rest or in a low motion state (determined by low variance of the accelerometer readings) for n_{rest} (currently, $n_{rest} = 120$ s), it assumes that the user is resting, and thus switches off all sensors for the next 60 s ($n_{rest}/2$ s). A 60 s time window for switching off all sensors ensures that *Annapurna* can still capture parts of an eating episode that starts during the n_{rest} period. Overall, from our feasibility study, we observed that choosing a low threshold for variance (less than 0.2 m/s^2 in a 1-min window) ensured that we did not miss any eating episode, while the day long data collection from the 3 participants indicated that on average this threshold could allow turning off all the sensors for $\approx$ 25 min in a day.
- As identified by Leech et al. [46], the inter-meal gap between two consecutive meals is at least 900 s. This indicates that once *Annapurna* detects the end of the meal, it can assume that the next meal will not occur within 900 s. *Annapurna* takes a conservative approach and currently turns off all sensors for $n_{b2b} = 600$ s, once it detects the end of one eating episode.
- If the smartwatch detects certain activities – e.g. walking or running for 2 consecutive minutes, it turns off all sensors for n_{act} seconds (currently, $n_{act} = 180$ s.), as it is highly unlikely for a user to have an in-plate meal while concurrently performing these activities. From the non-eating data that was collected from the 3 participants for an entire day we found that one participant's running session lasted for more than 30 min during this data collection. Additionally, as identified by Olzewski et al. in Singapore, 50% of individuals would use walking as their mode of transportation for a distance of 850 m [47]. Considering an average walking speed of 5 km/h, this translates to over 10 min of walking to cover the distance. Currently, we consider a society with an active lifestyle to design this heuristic. Further investigation for choosing the thresholds will be necessary for societies (or even individuals) with different active levels.
- As observed by Zhang et al. [25], the inter-chew gap (which directly translates to hand-to-mouth gestures) between two chewing bouts is less than 120 s in 80% cases. This indicates that if there is no hand-to-mouth gesture for 120 s (a hand-to-mouth gesture is the starting action in a chewing bout), it is highly likely that the eating episode has ended. Currently, if *Annapurna's Level 2* classifier does not observe 3 hand-to-mouth gestures in 300 s (60 additional seconds to minimize falsely missing gestures), then it switches back to the Level 1 classifier, thereby reducing energy overhead of running a gyroscope.

The final *Annapurna* system also utilizes human-in-the-loop principles to conserve energy and ensure accuracy. Whenever *Annapurna* determines the commencement of an eating episode, a pop up appears on the smartwatch to confirm if the individual is eating. If the user responded negatively to the pop up, *Annapurna* stopped the sensors and image capturing for the subsequent 300 s. If the user responded positively or does not respond at all, *Annapurna* assumes that the user is eating and carries on as it would have. Based on all the above mentioned heuristics, all sensors could be turned off for nearly 25% of the app running time in our dataset (details in Section 9). We have determined our heuristics and thresholds using data from participants who lead an active Singaporean lifestyle. Modification to these heuristics and thresholds might be necessary before deployment in other settings or cultures. Once an eating period is identified, the camera is triggered whenever *Annapurna* identifies a start of the subsequent hand-to-mouth gesture (from the output of the *Level2 Classifier*).

6. Image capturing technique

Once the eating gesture recognition module detects eating, the next step is to capture images using the camera. According to the Android Camera API documentation, the camera can capture either (a) videos or (b) images [48]. However, both these options were problematic for our use case:

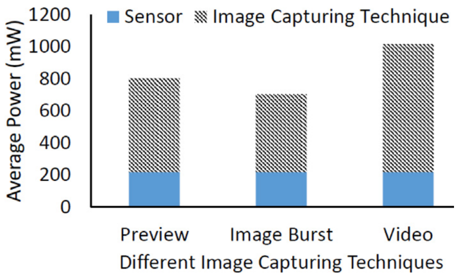


Fig. 8. Power consumption for image capturing.

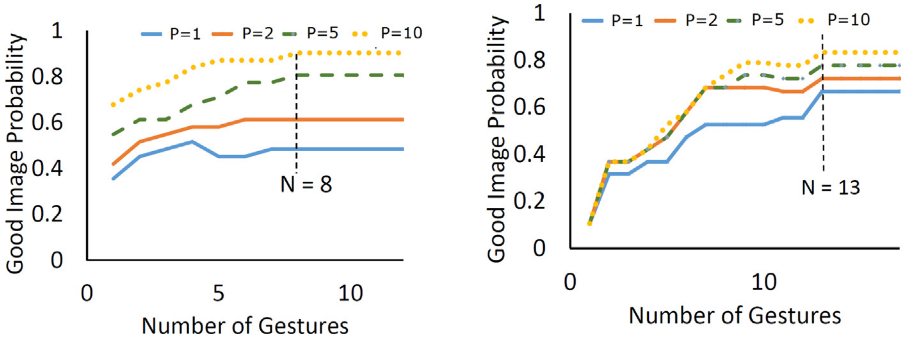


Fig. 9. Comparison of likelihood of acquiring a good image by the *Till-gesture* (left) and the *In-gesture* (right) strategies.

- *Capturing video*: Given the limited battery capacity of smartwatches, continuous video recording is not feasible, even if it is restricted to the eating episodes. By measuring the battery drain for continuous video capture, we found that the battery drains from 100% to 10% in ≈80 min.
- *Capturing image*: Capturing a single image reduces the probability of capturing images of the food item. We thus investigated capturing a burst of images. Capturing a burst of images has two issues: (i) latency – from triggering the camera to capturing a single image, the latency is ≈900 ms, and (ii) usability – as the number of images captured is small, the possibility of capturing an usable food plate image is extremely low.

From Android's API documentation we observed that Android also supports a *preview class*.

- *Preview class*: According to the API documentation, "*a camera preview class is a SurfaceView that can display the live image data coming from a camera, so users can frame and capture a picture or video*" [48]. This preview frame can be *grabbed* from the SurfaceView and stored. The preview frames refresh at a high rate (> 20 fps in the Gear 1), thus solving the latency issue that exists in capturing images. While of a lower quality, we found the quality of Preview frames to be good enough for image analysis.

To support low-energy continuous operation, we investigated the power consumption profile of the three modes. Fig. 8 shows the power consumption (measured using the Monsoon Power monitoring tool [49]) for different modes (along with inertial sensing). From the figure, we see that the *Burst* mode consumes the least power, while the *Preview* mode consumed only marginally higher power. Our feasibility studies showed that the burst mode could only capture an average of 2.7 images per gesture, while the preview mode captured 45.3 images per gesture (compared to 46.8 frames per gesture captured in video mode). Given our desire to capture a large set of images with low latency and low power consumption, the *Preview* mode is the most suitable approach.

Stopping the camera — Energy vs. Efficacy: An important question that now arises is — once the camera is triggered, how long should it be kept on? To answer this question, we computed the likelihood of getting at least one good image of the food content, in a ranked subset consisting of top p images from among all images captured till the Nth gesture, for each of the 135 eating episodes in the controlled-study (details of the ranking algorithm is explained in Section 7). If we considered the top 10 ranked images ($p = 10$), for a likelihood of 0.8 (80% of episodes had one good image), *Annapurna* had to wait for an average of 8 gestures and the camera sensor is kept on for approximately, 135 s. As an alternate strategy, we considered stopping the camera at the end of each gesture and starting it again at the next gesture (*In-gesture* strategy). In this case *Annapurna* had to wait for an average of 13 gestures and the camera sensor needs to be active for only approximately, 40 s. Even though the second approach kept the camera sensor ON for a shorter time duration, *Annapurna* adopted the former approach because (a) the complexity and the latency of triggering the camera

Table 5
Effectiveness of each image filtering step.

Filtering step (921617 Images)	Device	% of images remaining
Total images captured	Watch	100
RGB variance & Face filter	Phone	88
Edge count filter	Phone	37
Plate shape filter	Server	6.6
Depth map & CNN based Filter	Server	0.8

within the limited time of an ongoing individual gesture and (b) our desire to capture food item images even for smaller meals (ones with a lower number of eating gestures). Evaluation of both strategies for different values of p and N is shown in Fig. 9.

7. Image filtering

Our studies showed that many of the images captured by the preview mode were not *useful*–these included (i) blank images — when the camera captured only the table or other non-food items, (ii) blurry images — when the camera captured images while the hand was moving (iii) misleading images — when a neighbor's food plate or images with human faces was captured. We used a sequence of filters based on fairly standard image-processing operations to eliminate these irrelevant images. The key challenge was to balance the filtering efficiency (tradeoffs between false-positives and false-negatives) and the computational/energy overheads of different alternatives. We eventually adopted an architecture of 'lightweight early discard' (similar to the approach for continuous vision in Glimpse [50], where a relatively simple processing pipeline on the smartphone (to which the smartwatch is paired) eliminates a large percentage of low-quality images, and a second computationally-intensive pipeline on a backend server performs the ranking and selection of curated images.

7.1. Lightweight pre-processing on the phone

Images acquired by the smartwatch are transferred to the paired smartphone via Bluetooth. The phone in-turn is connected to a server over Wi-Fi. The obvious choice for performing the computationally intensive image processing operations (to filter irrelevant images) would be a server, rather than the smartphone. However, blindly transferring all the images to the server consumes both energy and bandwidth. To prevent *all* the images from being transferred to the server, *Annapurna* implements a lightweight pre-processor on the phone. The pre-processor utilizes (a) a color histogram based solid background detector, followed by (b) an initial face detection system using android's FaceDetector class, and (c) opencv's implementation of an edge detector. The background detector eliminates images with solid background (e.g. camera view blocked or pointing towards a wall/table), while the face detector eliminates images with visible human face. The edge detector filters images based on the observation that a clear image has a large number of edges, whereas blurry images have a smaller number of edges caused by motion blur.

By performing these simple image processing steps, we could eliminate 63% of the captured images even before the images were transferred to the server, thus saving both transmission power as well as bandwidth. Currently, we have investigated simple image processing steps on the smartphone. The simple image processing steps using OpenCV have little energy footprint on resource constrained devices [51]. Although more complex image processing operations can be performed on the smartwatch, the computation needs of these operations will introduce additional energy and latency overheads, but might reduce the communication overheads. With the advancement in deep learning techniques on resource constrained devices [52,53], in future, one might consider implementing more sophisticated deep learning techniques on either the smartphone or smartwatch. However, researchers should be wary about the energy drain of the deep learning approaches. As identified by Lane et al. the battery life of the device can drop to below 4 h when more complex deep learning approaches for image processing are implemented on mobile and wearable systems [17]. In future, it is necessary to investigate if such implementation can further optimize the image transfer, while ensuring that such techniques do not adversely affect the computation and battery life.

7.2. Detailed processing on the server

Images transferred to the server are processed to identify relevant images.

Determine shape of edge The first step at the server side processing is to identify if the edge is the edge of a plate. We assume that the plate has a regular, convex shape (either rectangular or circular). To determine rectangular shape, we identify straight lines whose length is above a minimum threshold. Similarly, we use the *approxPolyDP* function in *opencv* to compute the number of curves in the edge. If the number of curves is above a threshold, and the slopes of the consecutive curves indicate a monotonic increase or decrease, the overall shape is convex and a candidate for a plate's outline.

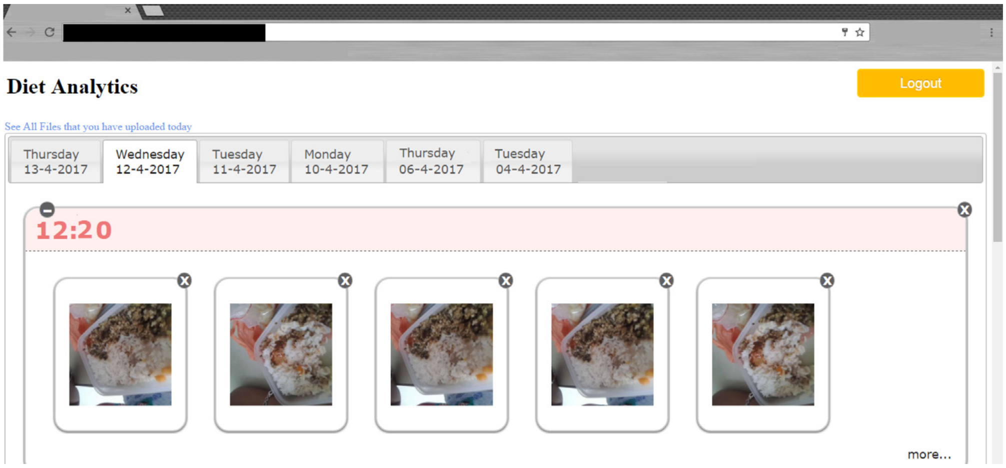


Fig. 10. View of the *Annapurna* portal.

Eliminating using a depth map Several irrelevant images were observed to contain edges, but from objects (e.g., pictures on the wall, or from the neighbor's plate) that were distinct from the user's food container. To eliminate such images, a depth map is constructed (via the parallax method) from the acquired sequence of images. The dominant SURF features [54] in two images, taken 300 ms apart, are identified. The pixel disparity (after common axis alignment) between the identified features is evaluated to build the depth map: foreground objects have higher disparity than background objects. If the rectangular/circular object detected in the image is in the foreground, then this image is saved as a likely image of the food plate; else, it is discarded.

Non-food elimination via deep learning Although the aforementioned steps help in eliminating irrelevant images, there is still uncertainty about whether the image contains food images. To ensure that the image is indeed that of a food item, we then invoke the API provided by Clarifai inc. [55]. This API utilizes CNN to identify the likelihood of presence of food in an image.

Finally, all images that pass these filtering steps are stored, and ranked based on a 'visibility area' score: this score is directly proportional to the area of extrapolated rectangle. Table 5 shows the average fraction of images that were eliminated at each step on our controlled study corpus.

8. Building the food journal

We next identify a users' food journaling preferences.

8.1. *Image preference (survey)*

We surveyed 32 students and researchers (16 male and 16 female) to understand (a) how many food images they would like to see per eating episode, and (b) how sensitive they were to erroneous images taken by *Annapurna* (e.g., images that did include the food or were too blurry).

Overall, 70% of the respondents indicated that at least 5–10 food images were appropriate per eating episode. We suspect this is because users would like to see the food they ate from different angles or they did not want episodes where no correct food images were shown. Moreover, 80% of the respondents indicated they would still use the system (16% definitely, 37% probably and 28% possibly) if at least 80% of the displayed images were correct food images.

8.2. *Annapurna web portal*

Annapurna's web portal is shown in Fig. 10. The portal is configured to display the top-5 images (determined by the image processing algorithm) to the user. Our food image selection algorithm has an 88% accuracy at picking correct food images (details in Section 8.3). In a day-wise tabbed view users can see photos of the food consumed on any particular day. For each day, the food photos are separated into different meals based on the time of consumption.

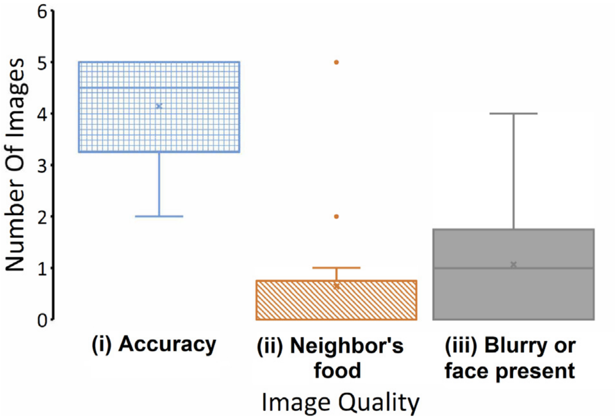


Fig. 11. Assessment of image quality. The plot shows (i) the accuracy with which *Annapurna* could identify images of food, as displayed in the portal (blue), (ii) number of images which had images of food items that were not consumed by the individual (orange), and (iii) images that were either blurry or captured someone's face (gray).

8.3. *Effectiveness of the food journal*

To evaluate the effectiveness of the food journal and its image ranking algorithm, we conducted a semi-controlled free-living study with 14 participants (4 females, 10 males, aged between 23 and 36). Participants were instructed to consume one meal while wearing a smartwatch (with our app running). We did not provide any instructions about where, when, what, or how to eat their food.

The participants consumed their meals in 7 distinct locations (2 public seating areas, 3 food courts, and 2 restaurants); consuming a diverse food item choice including buns, rice cakes, rice with curry, pasta, and salad. The food items were consumed using some combination of their hands (2 episodes), spoons (8 episodes), forks (7 episodes), and chopsticks (2 episodes).

Results: For each participant, *Annapurna* journal displayed the top 5 images. The participants were then asked to evaluate the system in terms of (1) image accuracy — did the images contain food items that they consumed, and (2) image quality — were the images clear? Fig. 11 shows the results for image accuracy and quality, computed over each set of 5 images representing one meal episode. The Y axis of the figure indicates the number of images.

8.3.1. *Image accuracy*

The *Annapurna* journal displayed 4.14 valid food images (out of 5) on average. We define a valid food image as one that shows at least one food item that the participant consumed. For 7 participants, all the 5 displayed images were valid. Also, there was no eating episode where food images did not get captured at all. The worst result had just 2 valid food images and occurred for a participant consuming rice cakes with relatively little hand movement.

8.3.2. *Image quality*

Overall, the participants could clearly identify about 80% of the displayed pictures, with only 1.07 images considered blurry on average. In the worst case, one participant marked 3 of the 5 images as blurry, although even this person noted that he could still identify the food items in the images. This observed sensitivity to such motion blur led us to increase *Annapurna's* blur threshold in the final production-ready version.

9. In-the-wild evaluation

The semi-controlled study allowed us to validate the feasibility of capturing images of food items consumed by participants in an uncontrolled environment, during explicit eating episodes. As a logical next step, we explore the end-to-end feasibility of *Annapurna* as an automated food journal, which tracks a user's eating episodes throughout the day, while the user continues with their everyday activities. To this end, we recruited participants to wear the smartwatch continuously throughout the day. Overall, we conducted two free-living studies with a stable *eating gesture recognizer* module – *uncontrolled free-living Study 1 (S1)* with 4 participants (1 male, 3 females) and *uncontrolled free-living Study 2 (S2)* with 7 participants (3 females, 4 males). *S1* allowed us to evaluate the feasibility of capturing eating gestures in free-living settings. The eating recognizer module described in Section 5.3 was used in *S1*. However, the battery life of the smartwatch was poor. We thus incorporated several refinements to the eating recognizer module (as described in Section 5.4) and that was used in *S2*. Table 6 provides extensive details about the two studies. In the table, *TP* (true positive) indicates eating episodes that were correctly identified and displayed to the user, *FP* (false positive) refers to

Table 6
Performance of *Annapurna* in the completely free-living studies.

		Uncontrolled free-living study 1 (S1)				Uncontrolled free-living study 2 (S2)					Overall
Ground truth	Participant Id	1	2	3	4	4	5	6	7	8	9
	# days	5	5	5	5	6	5	5	4	6	46
	App time (h)	–	–	–	–	33.7	26.1	32.6	26	33.9	152.3
	# Meals (GT)	7	7	6	10	11	9	11	8	12	81
Performance	TP	7	7	6	9	11	7	11	7	12	77
	FP-Inertial	2	2	1	4	7	7	7	5	10	45
	FP-Overall	1	0	0	1	1	1	0	0	0	4
	FN	0	0	0	1	0	2	0	1	0	4
	FNs - Camera	0	0	0	0	0	1	0	1	0	2
Sensor use (hours)	All Sensors OFF					11.7	5.4	8.3	3.9	8	37.3
	Accel ON					13.6	14.9	19.1	14.9	18.6	81.2
	Accel + Gyro ON		N/A			7.8	5.3	4.6	6.8	6.5	31
	Camera ON					0.6	0.5	0.6	0.4	0.8	2.9

TP: True positive (Correct meal detection), *FP-inertial*: False positives triggered by gesture recognizer, *FP-Overall*: False positives in the final journal after image filtering, *FN*: False negative (Meals missed by App), *FNs Camera*: FN due to Camera triggering failure.
Participant # 4 participated in both studies.

episodes that our system falsely inferred as eating, and *FN* indicates eating episodes that we missed. Note that even if the gesture recognition system falsely identified eating episodes, the image filtering technique was successful in removing these *FP* (*FP-Overall*).

9.1. S1 : Uncontrolled free-living study 1

4 participants were recruited and were provided with a smartwatch and a smartphone with the *Annapurna* app installed. They were asked to appropriately recharge the battery. While they had no directives in terms of their eating decisions (when, where and what to eat), the participants were instructed to validate the images uploaded to *Annapurna* Web Journal at the end of the day. In addition to validating the images in food journal, the participants were also instructed to go through all the images that were uploaded to the server. In order to protect their privacy, they were instructed to delete any image which captured sensitive context and they were not comfortable in sharing. The research team accessed the uploaded images only after the participant indicated that they were okay with sharing the uploaded images.

The 4 participants wore the smartwatch for 5 days each. During this study, *Annapurna* recalled 29 eating episodes reported by the users, while missing just one eating episode. Also there were just 2 false positives (when *Annapurna* presented images from non-eating periods in the journal) across all users. On a closer look, there were 9 times when the gesture recognizer had actually triggered the camera incorrectly. This included 2 episodes where the participant consumed only a beverage. However, the images acquired from 7 of these cases, including the 2 only beverage episodes were rejected by the image filters.

9.2. S2: Uncontrolled free-living study 2

At the end of S1, we surveyed the participants (from S1 and also from the previous pilot study) about the usability of the system. A total of 6 *Annapurna* users responded back. The users indicated improvements that they would like in future version: (a) 3 users wanted a mechanism to automatically compute the calories consumed, (b) 1 user suggested that we should display the total duration of every meal, and (c) 1 user wanted to manually inspect the images captured before they were sent to the server. While (a) is outside our scope, *Annapurna* can be easily modified to support (b) and (c).

We recruited 7 participants for the subsequent study with a more power efficient app. However, 2 participants had issues with the app running on their watch, and hence could not be effectively included in the study. We followed the same protocol as S1 and collected 26 person days of data from the 5 participants, which resulted in 152.3 h of data collection. During the 152.3 h, 51 meals were consumed by the participants while they wore the smartwatch. Since *Annapurna* turned off all sensors after 8 eating gestures were detected, we could not measure the duration of the meals in the free-living setting, Among these 51 meals, *Annapurna* could correctly identify 48 meals, indicating that the energy-saving heuristics did not affect the recall of the system. In terms of false positives, 36 false positive (*FP*) episodes were detected by the gesture recognition module. This included 5 episodes where the participant consumed only a beverage. In all, 34 among these 36 episodes were eliminated by the image filtering module, including all five only-beverage episodes (*FP-Overall*= 2). The images which were unfiltered had closeup images of objects that resembled food items.

By measuring the power consumption of each of the sensors, we had found that the accel alone consumed ≈20 mW power, accel + gyro together consumed ≈150 mW power and the camera along with the inertial sensors consumed ≈800 mW power. Overall, we found that based on the simple heuristics, we could turn off all sensing for ≈24.5% of the entire application running time without any significant drop in accuracy. Additionally, the gyroscope ran for less than 21% of the total app running time, thus allowing longer battery life.

9.3. Overall Performance of Annapurna

From S1 and S2 we observed that *Annapurna* could recall 95% meals (77 out of 81) that were consumed by the participants. Although gesture recognition itself had several false positives, the precision of the system was 95% after the image filtering (compared to 63% precision in the absence of such filtering). The energy saving heuristics assisted in saving nearly 25% of *Annapurna's* sensing time, while the app had expensive sensors (e.g., camera and gyroscope) disabled for approx. 80% of a typical day.

10. Discussion and future work

While *Annapurna* affirms the possibility of building an automated food journaling system, in future, several additional issues need to be explored.

Large scale user study: We have currently tested *Annapurna* with 8 unique participants in multiple free living studies, across a total of 46 days (an average of 5 days per participant). The duration is definitely longer and more rigorous as compared to several existing free living eating activity monitoring studies [56]. However, it must be noted that the current study sample size is small for claiming generalizability of the system. The studies have currently been conducted with participants who are affiliated with an university in Singapore. The eating style in Singapore might differ from eating styles in other locations, although it appears to be broadly representative of eating styles across Eastern and Southern Asia. A free living study in another geographic location might help in understanding the general eating pattern. Currently, *Annapurna* has been tested on a healthy population, who are in the normal BMI range. In future, it will be useful to test *Annapurna* with a clinically vulnerable population to understand how an automated food journaling system can help such population.

Identifying food items and estimating calorific values: Currently, *Annapurna* captures usable food-images but does not automatically identify the food item and an estimate of its calorific content. Work by Kawano et al. attests to the possibility of such automatic food identification [37], while more recently, Sahoo et al. have applied deep learning-based approaches for automated food recognition [42]. However, these techniques utilize a clear, *frontal* image of the food item, and further investigation is necessary to determine whether they will work on *Annapurna's* food images, which have partial, sometimes blurry, views of the food item. Orthogonally, Dong et al. have observed a correlation between the number of hand to mouth gestures and caloric intake [23]. *Annapurna* can use a similar approach for caloric estimate in the future.

Using in-situ deep learning approaches: As discussed in Section 7, *Annapurna* currently leverages on simple image processing steps on the smartphone, before transferring the images to the server for further analysis. With advancement in on-device deep learning implementation on resource constrained devices [52,53,57] and improvement in capabilities of modern smartwatches, deploying these deep learning techniques on smartwatches and smartphones is gradually becoming a possibility. Such implementation will reduce the communication overhead for the smartwatch. In case deep learning approach is implemented on a smartphone, it might further optimize the image filtering process. However, as discussed in Section 7, *Annapurna* currently implements simple and low-energy computation. Further communication and computation trade-off analysis is necessary before using deep learning approaches on the smartwatch to optimize the processing pipeline.

Alternate approaches: Currently, *Annapurna* utilizes a smartwatch-based camera to capture an image of the food item. Alternately, we can explore other wearable devices for such automated image capture. For example, a smartglass-mounted camera, such as one proposed by Zhang et al. [28] or camera as used by Liu et al. [3] is likely to obtain a clearer image of the food being consumed. Similar to *Annapurna's* current implementation, the gesture recognizer module running on a smartwatch can trigger the camera on a different device. Such approaches, however, require multiple wearable devices and tighter synchronization across devices. Additionally, one must keep inter-device communication latency in mind while implementing such an approach.

Capturing additional facets: Annapurna currently captures a small subset of representative images of the food item: for energy efficiency, the image capture is disabled, once *Annapurna* determines that an appropriate number of images has been recorded. However, we may be interesting in automatically tracking the total amount of food consumed. This can be done, for example, by taking an image of the plate at the *end* of a meal, and comparing the size of the observed food item with a corresponding image taken at the beginning of the meal. This will require changes to *Annapurna's* image capturing logic, and would impose a different energy-vs-accuracy operational trade-off.

Ensuring user privacy To detect the food plate, *Annapurna* automatically captures images using the smartwatch's RGB camera. However, in addition to images of the food plate, false positives in the gesture recognition pipeline leads to capturing unnecessary and sometimes sensitive context. In our study, at the end of the day, participants manually reviewed each uploaded image to ensure that the captured images were not privacy encroaching. This allowed preserving user privacy in the user studies. Currently, *Annapurna* does not apply any image obfuscation technique on the images, i.e., capturing only the food plate, while removing any sensitive context. Recently, there has been substantial advancement in thermal imaging approaches for food item detection [30,58], and for obfuscation of unnecessary context [59]. Infrared sensor arrays such as the GRID-EYE [60] consume little power and provide thermal details of the scene. In future, we can consider augmenting the output of the RGB camera with the thermal sensor to allow detecting food items in scene and to enhance user's privacy by obfuscating unnecessary (and possibly privacy-sensitive) context.

11. Conclusion

We presented *Annapurna*, the first known implementation of an end-to-end automated smartwatch-based food journaling system. *Annapurna* first identifies eating gestures, and then capture images of the food item by activating the camera-sensor during the eating activity. We observed that a smartwatch-embedded camera can indeed capture such food-related images, with the preview mode providing the balance between energy efficiency and image relevance. Through image filtering and processing, we demonstrated that *Annapurna* could identify the correct images in over 80% cases. Finally, through multiple free-living studies in Singapore we showed that *Annapurna* could achieve a precision and recall of 95% in naturalistic environments, even when either no sensor or a low-energy accelerometer sensor was running during $\approx$ 80% of the detection duration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported partially by Singapore Ministry of Education Academic Research Fund Tier 2 under research grant MOE2014-T2-1063, and by the National Research Foundation, Prime Minister's Office, Singapore under its IDMFutures Funding Initiative. All findings and recommendations are those of the authors and do not necessarily reflect the views of the granting agency, or SMU.

References

[1] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, M. Hansen, Image browsing, processing, and clustering for participatory sensing: lessons from a DietSense prototype, in: Proceedings of the Workshop on Embedded Networked Sensors, 2007, pp. 13–17.

[2] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, K. Wood, SenseCam: A retrospective memory aid, in: International Conference on Ubiquitous Computing, 2006, pp. 177–193.

[3] J. Liu, E. Johns, L. Atallah, C. Pettitt, B. Lo, G. Frost, G.-Z. Yang, An intelligent food-intake monitoring system using wearable sensors, in: Conference on Wearable and Implantable Body Sensor Networks, IEEE, 2012, pp. 154–160, http://dx.doi.org/10.1109/bsn.2012.11.

[4] S. Bi, K. Caine, R. Halter, J. Sorber, D. Kotz, T. Wang, N. Tobias, J. Nordrum, S. Wang, G. Halvorsen, S. Sen, R. Peterson, K. Odame, Auracle: Detecting eating episodes with an ear-mounted sensor, Proc. ACM Interact. Mobile Wearable Ubiquitous Technol. 2 (3) (2018) 1–27, http://dx.doi.org/10.1145/3264902.

[5] A. Bedri, R. Li, M. Haynes, R.P. Kosaraju, I. Grover, T. Prioleau, M.Y. Beh, M. Goel, T. Starner, G. Abowd, EarBit: Using wearable sensors to detect eating episodes in unconstrained environments, Proc. ACM Interact. Mobile Wearable Ubiquitous Technol. 1 (3) (2017) 1–20, http://dx.doi.org/10.1145/3130902.

[6] K. Yatani, K.N. Truong, BodyScope: a wearable acoustic sensor for activity recognition, in: ACM Conference on Ubiquitous Computing, in: Ubicomp'12, 2012, http://dx.doi.org/10.1145/2370216.2370269.

[7] T. Rahman, A.T. Adams, M. Zhang, E. Cherry, B. Zhou, H. Peng, T. Choudhury, Bodybeat: A mobile system for sensing non-speech body sounds, in: Annual International Conference on Mobile Systems, Applications, and Services, in: MobiSys'14, 2014, http://dx.doi.org/10.1145/2594368.2594386.

[8] K.S. Chun, S. Bhattacharya, E. Thomaz, Detecting eating episodes by tracking jawbone movements with a non-contact wearable sensor, Proc. ACM Interact. Mobile Wearable Ubiquitous Technol. 2 (1) (2018) 1–21, http://dx.doi.org/10.1145/3191736.

[9] E. Thomaz, I. Essa, G.D. Abowd, A practical approach for recognizing eating moments with wrist-mounted inertial sensing, in: Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, in: UbiComp '15, 2015.

[10] S. Zhang, W. Stogin, N. Alshurafa, I sense overeating: Motif-based machine learning framework to detect overeating using wrist-worn sensing, Inf. Fusion 41 (2018) 37–47.

[11] Narrative clip, 2019, http://getnarrative.com/, accessed: 2019-12-01.

[12] Food AI, 2019, https://foodai.org/, accessed: 2019-12-01.

[13] L. Deng, J. Chen, Q. Sun, X. He, S. Tang, Z. Ming, Y. Zhang, T.S. Chua, Mixed-dish recognition with contextual relation networks, in: ACM International Conference on Multimedia, Association for Computing Machinery, Inc, 2019, pp. 112–120, http://dx.doi.org/10.1145/3343031.3351147.

[14] N.I. Lipoeto, K.G. Lin, I. Angeles-Agdeppa, Food consumption patterns and nutrition transition in South-East Asia, Public Health Nutr. 16 (9) (2013) 1637–1643, http://dx.doi.org/10.1017/S1368980012004569.

[15] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, P. Bahl, Energy characterization and optimization of image sensing toward continuous mobile vision, in: Proceeding of the Annual International Conference on Mobile Systems, Applications, and Services, 2013, pp. 69–82.

[16] S. Sen, V. Subbaraju, A. Misra, R.K. Balan, Y. Lee, The case for smartwatch-based diet monitoring, in: International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), 2015, pp. 585–590.

[17] N.D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, F. Kawsar, An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices, in: International Workshop on Internet of Things Towards Applications, 2015.

[18] MyFitnessPal, 2019, https://www.myfitnesspal.com/, accessed: 2019-12-01.

[19] B.L. Heitmann, L. Lissner, Dietary underreporting by obese individuals–is it specific or non-specific? Bmj 311 (7011) (1995) 986–989.

[20] S.D. Poppitt, D. Swann, A.E. Black, A.M. Prentice, Assessment of selective under-reporting of food intake by both obese and non-obese women in a metabolic facility, Int. J. Obes. Relat. Metab. Disord. 22 (4) (1998).

[21] S. Cadavid, M. Abdel-Mottaleb, A. Helal, Exploiting visual quasi-periodicity for real-time chewing event detection using active appearance models and support vector machines, Pers. Ubiquitous Comput. 16 (6) (2012) 729–739.

[22] B. Zhou, J. Cheng, M. Sundholm, A. Reiss, W. Huang, O. Amft, P. Lukowicz, Smart table surface: A novel approach to pervasive dining monitoring, in: International Conference on Pervasive Computing and Communications, IEEE, 2015, pp. 155–162.

[23] Y. Dong, A. Hoover, J. Scisco, E. Muth, A new method for measuring meal intake in humans via automated wrist motion tracking, Appl. Psychophysiol. Biofeedback 37 (3) (2012) 205–215.

[24] S. Sen, V. Subbaraju, A. Misra, R. Balan, Y. Lee, Annapurna: Building a real-world smartwatch-based automated food journal, in: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, in: WoWMoM, 2018.

[25] S. Zhang, Y. Zhao, D.T. Nguyen, R. Xu, S. Sen, J. Hester, N. Alshurafa, Necksense: A multi-sensor necklace for detecting eating activities in free-living conditions, Proc. ACM Interact. Mobile Wearable Ubiquitous Technol. 4 (2) (2020) 1–26, http://dx.doi.org/10.1145/3397313.

[26] A. Bedri, D. Li, R. Khurana, K. Bhuwalka, M. Goel, Fitbyte: Automatic diet monitoring in unconstrained situations using multimodal sensing on eyeglasses, in: Conference on Human Factors in Computing Systems, Vol. 20, 2020, pp. 1–12, http://dx.doi.org/10.1145/3313831.3376869.

[27] R. Zhang, O. Amft, Free-living eating event spotting using EMG-monitoring eyeglasses, in: International Conference on Biomedical and Health Informatics, BHI 2018, Vol. 2018-Janua, 2018, pp. 128–132, http://dx.doi.org/10.1109/BHI.2018.8333386.

[28] R. Zhang, O. Amft, Monitoring chewing and eating in free-living using smart eyeglasses, IEEE J. Biomed. Health Inf. 22 (1) (2018) 23–32, http://dx.doi.org/10.1109/JBHI.2017.2698523.

[29] M. Mirtchouk, C. Merck, S. Kleinberg, Automated estimation of food type and amount consumed from body-worn audio and motion sensors, in: ACM International Joint Conference on Pervasive and Ubiquitous Computing, in: UbiComp, ACM, New York, NY, USA, 2016, pp. 451–462, http://dx.doi.org/10.1145/2971648.2971677.

[30] J. Lee, A. Banerjee, S.K.S. Gupta, MT-Diet: Automated smartphone based diet assessment with infrared images, in: Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on, IEEE, 2016, pp. 1–6.

[31] J. Noronha, E. Hysen, H. Zhang, K.Z. Gajos, PlateMate: Crowdsourcing nutrition analysis from food photographs, in: ACM Symposium on User Interface Software and Technology, ACM Press, 2011, pp. 1–11, http://dx.doi.org/10.1145/2047196.2047198.

[32] O. Amft, H. Junker, G. Troster, Detection of eating and drinking arm gestures using inertial body-worn sensors, in: International Symposium on Wearable Computers, 2005, pp. 160–163.

[33] X. Ye, G. Chen, Y. Cao, Automatic eating detection using head-mount and wrist-worn accelerometers, in: International Conference on E-Health Networking, Application & Services, in: HealthCom, IEEE, 2015, pp. 578–581.

[34] Y. Dong, J. Scisco, M. Wilson, E. Muth, A. Hoover, Detecting periods of eating during free-living by tracking wrist motion, J. Biomed. Health Inform. 18 (4) (2014) 1253–1260, http://dx.doi.org/10.1109/JBHI.2013.2282471.

[35] E. Thomaz, A. Parnami, J. Bidwell, I. Essa, G.D. Abowd, Technological approaches for addressing privacy concerns when recognizing eating behaviors with wearable cameras, in: ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2013, pp. 739–748, http://dx.doi.org/10.1145/2493432.2493509.

[36] F. Zhu, M. Bosch, C.J. Boushey, E.J. Delp, An image analysis system for dietary assessment and evaluation, in: International Conference on Image Processing, 2010, pp. 1853–1856.

[37] Y. Kawano, K. Yanai, Foodcam: A real-time food recognition system on a smartphone, Multimedia Tools Appl. 74 (14) (2015) 5263–5287.

[38] P.V. Rouast, M.T. Adam, Learning deep representations for video-based intake gesture detection, IEEE J. Biomed. Health Inf. 24 (6) (2020) 1727–1737, http://dx.doi.org/10.1109/JBHI.2019.2942845.

[39] D. Hossain, T. Ghosh, E. Sazonov, Automatic count of bites and chews from videos of eating episodes, IEEE Access 8 (2020) 101934–101945, http://dx.doi.org/10.1109/ACCESS.2020.2998716.

[40] G. O'Loughlin, S.J. Cullen, A. McGoldrick, S. O'Connor, R. Blain, S. O'Malley, G.D. Warrington, Using a wearable camera to increase the accuracy of dietary analysis, Am. J. Prev. Med. 44 (3) (2013) 297–301.

[41] M. Bolanos, P. Radeva, Simultaneous food localization and recognition, in: IEEE International Conference on Pattern Recognition (ICPR), 2016.

[42] D. Sahoo, W. Hao, S. Ke, W. Xiongwei, H. Le, P. Achananuparp, E.-P. Lim, S.C. Hoi, FoodAI: Food image recognition via deep learning for smart food logging, in: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2019.

[43] M. Corbalan-Tutau, J. Madrid, M. Garaulet, Timing and duration of sleep and meals in obese and normal weight women. Association with increase blood pressure, Appetite 59 (2012) 9–16.

[44] K. Ang, S. Foo, An exploratory study of eating patterns of Singapore children and teenagers, Health Manage. 102 (2002) 239–248.

[45] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, K. Aberer, Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach, in: International Symposium on Wearable Computers, 2012, pp. 17–24.

[46] R.M. Leech, A. Worsley, A. Timperio, S.A. McNaughton, Characterizing eating patterns: a comparison of eating occasion definitions, Am. J. Clin. Nutr. 102 (5) (2015) 1229–1237, http://dx.doi.org/10.3945/ajcn.115.114660.

[47] P. Olzewski, C. Tan, Walking lessons: Pedestrian travel in Singapore, Traffic Eng. Control 40 (10) (1999).

[48] Android camera API, 2019, https://developer.android.com/guide/topics/media/camera, accessed: 2019-12-01.

[49] Monsoon power monitor, 2019, https://www.msoon.com/, accessed: 2019-12-01.

[50] S. Naderiparizi, P. Zhang, M. Philipose, B. Priyantha, D. Ganesan, J. Liu, Glimpse: A programmable early-discard camera architecture for continuous mobile vision, in: Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services, 2017.

[51] A. Anuar, K.M. Saipullah, N.A. Ismail, Y. Soo, Opencv based real-time video processing using android smartphone, Int. J. Comput. Technol. Electron. Eng. (IJCTEE) 1 (3) (2011) 1–6.

[52] S. Bhattacharya, N.D. Lane, From smart to deep: Robust activity recognition on smartwatches using deep learning, in: 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), 2016, pp. 1–6, http://dx.doi.org/10.1109/PERCOMW.2016.7457169.

[53] N.D. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi, F. Kawsar, Squeezing deep learning into mobile and embedded devices, IEEE Pervasive Comput. 16 (3) (2017) 82–88, http://dx.doi.org/10.1109/MPRV.2017.2940968.

[54] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, SURF: Speeded up robust features, in: European Conference on Computer Vision, 2006, pp. 404–417, http://dx.doi.org/10.1007/11744023{_}32.

[55] Clarifai, 2019, https://www.clarifai.com/, accessed: 2019-12-01.

[56] B.M. Bell, R. Alam, N. Alshurafa, E. Thomaz, A.S. Mondol, K. de la Haye, J.A. Stankovic, J. Lach, D. Spruijt-Metz, Automatic, wearable-based, in-field eating detection approaches for public health research: a scoping review, NPJ Digit. Med. 3 (1) (2020) http://dx.doi.org/10.1038/s41746-020-0246-2.

[57] C.R. Banbury, V.J. Reddi, M. Lam, W. Fu, A. Fazel, J. Holleman, X. Huang, R. Hurtado, D. Kanter, A. Lokhmotov, D. Patterson, D. Pau, J.-s. Seo, J. Sieracki, U. Thakker, M. Verhelst, P. Yadav, Benchmarking TinyML systems: Challenges and direction, 2020, Arxiv preprint arXiv:2003.04821.

[58] A. Sharma, A. Misra, V. Subramaniam, Y. Lee, SmrtFridge: IoT-based, user interaction-driven food item & quantity sensing, in: Conference on Embedded Networked Sensor Systems, Association for Computing Machinery, Inc, 2019, pp. 245–257, http://dx.doi.org/10.1145/3356250.3360028.

[59] R. Alharbi, M. Tolba, L.C. Petito, J. Hester, N. Alshurafa, To mask or not to mask? Balancing privacy with visual confirmation utility in activity-oriented wearable cameras, Proc. ACM Interact. Mobile Wearable Ubiquitous Technol. 3 (3) (2019) 1–29, http://dx.doi.org/10.1145/3351230.

[60] Grid-EYE® infrared array sensor | panasonic industrial devices, 2020, https://na.industrial.panasonic.com/products/sensors/sensors-automotive-industrial-applications/lineup/grid-eye-infrared-array-sensor, accessed: 2020-05-20.