Annapurna: Building a Real-World Smartwatch-based Automated Food Journal

Sougata Sen, Vigneshwaran Subbaraju, Archan Misra, Rajesh Balan, Youngki Lee
School of Information Systems, Singapore Management University
{sougata.sen.2012, vigneshwaran, archanm, rajesh, youngkilee}@smu.edu.sg

***Abstract*—We describe the design and implementation of a smartwatch-based, completely unobtrusive, food journaling system, where the smartwatch helps to intelligently capture useful images of food that an individual consumes throughout the day. The overall system, called *Annapurna*, is based on three key components: (a) a smartwatch-based gesture recognizer to identify eating gestures, (b) a smartwatch-based image capturer that obtains a small set of relevant and useful images with a low energy overhead, and (c) a server-based image filtering engine that removes irrelevant uploaded images, and then catalogs them through a portal. Our primary challenge is to make the system robust to the huge diversity in natural eating habits and food choices. We show how we address this by an appropriate coupling between a smartwatch's camera sensor and inertial sensor-based tracking of eating gestures, thereby helping to capture multiple likely-to-be-useful images with low energy overhead. Through a series of real-world, in-the-wild studies, we demonstrate the end-to-end working of *Annapurna*, which captures useful images in over 95% of all natural eating episodes.**

I. 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 end goals–e.g., losing or maintaining target weight, or capturing unhealthy habits, such as eating fast or having meals late at night. Most of the proposed solutions are either obtrusive (e.g., phone hanging around neck [1] or clipped to clothing [2]), or rely on specialized wearable sensors (e.g. [3], [4]). More recently, researchers have utilized the inertial sensors on a more mainstream wearable to capture the 'intake' gesture during eating [5]. Such wrist-worn approaches hold promise as an unobtrusive, automated way to identify eating episodes, but don't capture images of the food being consumed–i.e., do not help answer the question "What did you eat?".

In our work, we take the concept of automatic food-journaling, using a commodity wrist-worn wearable device (e.g., a smartwatch or a smartband), a step further: we demonstrate how to use a camera mounted on such a wearable to capture & curate *useful* images of the food items consumed. Our core idea is simple: (a) a smartwatch's inertial sensors help identify the eating-related "hand-to-mouth" gestures; while (b) the embedded camera is then clevery triggered to take appropriate pictures, when it has a clear, unobstructed view of the food being consumed. In this work, we do not focus too much on the gesture recognition component (relevant details can be found in our earlier work [6], [7]). Instead, we focus on the techniques for using the embedded camera to automatically and opportunistically capture and curate the image of the food itself efficiently. This *opportunistic image-capture* approach simultaneously addresses the energy efficiency and ethical concerns [8] by minimizing unrelated image-capture.

978-1-5386-4725-7/18/$31.00 2018 IEEE

A practical embodiment of this "simple idea" has four key principal challenges: (a) *Can a smartwatch camera capture meaningful images of the food being eaten?* This will not only depend on the position of the smartwatch's camera, but also on the individual's eating style and mode (e.g., forks, spoon or chopsticks); (b) *Can such gesture-driven image capture work across the diversity of real-world food choices and eating environments?* In the real world, many factors affect eating gestures, including the food type and the environment; (c) *Can we capture useful images in an energy-efficient fashion?* A naive approach, of activating video recording continuously, has prohibitively high energy overheads; and (d) *Can we automatically filter out spurious images?* To both reduce transmission energy costs and assuage privacy concerns, we should be able to select only relevant images. Moreover, this selection pipeline should ideally be distributed, applying the principle of "early discard" [9] closer to the edge device.

In this paper, we design and evaluate a system that addresses the above challenges. This enables the development of *Annapurna*, a smart-watch application which can automatically identify eating periods, trigger the camera at the right time to capture images of the food plate in a energy efficient manner, identify relevant images and finally present the relevant images as a food journal to the end user. Our design is driven by extensive real-world observations of eating behavior and validated through a series of studies: (*Study 1*) is a controlled study involving 21 individuals consuming 5 different food types, (*Study 2*) a progressively evolving in-the-wild study involving 7 users (4 user's 20 days of data has been reported in this paper), and (*Study 3*) another in-the-wild study with additional system-level refinements for improved energy efficiency and accuracy, with 5 users. *Study 1* established the feasibility of using a smartwatch camera to capture food images, *Study 2* showed the overall working of *Annapurna* in real-world conditions, and *Study 3* established that heuristics could help reduce the overall sensing energy overhead without compromising on accuracy.

Key Contributions: With gesture recognition being covered in [7], this paper answers the following key questions:

- *Can Useful Images be Captured?* In Section III, we show

that the answer is '*yes*' (we can capture such an image in 90%+ episodes for common food items, and in 65%+ cases for more unusual choices). However, this success depends on the on-watch position of the camera.

- *Can the Process of Image Capture be Optimized?* In Section VI, we show how the *Preview* mode achieves the best balance between the goals of minimizing energy-intensive 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 msec) latency.
- *Can standard image processing techniques identify food items in thee images?* in Section VII, we demonstrate how a pipeline of well-known, computationally-simple image processing techniques can eliminate irrelevant images.
- *Does the overall system work in the real world?* Through multiple in-the-wild studies collected over 46 person-days (described in Section VIII), we demonstrate that *Annapurna* could capture in-the-wild eating episodes with a precision and recall of 95% each.

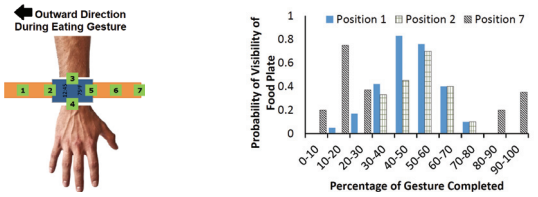
II. Related Work

Recognizing eating related gestures via inertial sensors on wrist-worn devices has been extensively studied [5], [7], [10], [11]. However, the possibility of food journaling by triggering the camera on the smartwatch has not been investigated. In this section we discuss about existing food journaling systems.

Food journals based on manual entry (e.g. [12]) allow users to track the food they consumed; however, research shows that people tend to under-report food consumed [13]. More recently, researchers have explored the images captured by individuals to identify the food consumed ([14]). However, they are also plagued by similar problems associated with manual effort since they require the user to explicitly acquire the images of the food.

Fully-automated systems for food journaling with images acquired from wearables has been explored previously ([1]–[3]). In [1], the authors use a phone hung around the neck to periodically capture ego-centric images of the food. Similarly, a wearable life-logging device is used in [2]. Functionally, the closest related work to this paper is by Liu et al. [3], where the authors use a specialized hardware (microphone and camera) to detect the chewing sound using the microphone and subsequently triggers the camera to capture images of the food being consumed. In contrast, in our work, we use a commercial smartwatch which is readily available and more conveniently adopted as compared to the specialized hardware.

The above works [1]–[3] have also discussed the possibility of filtering the relevant food-related images using standard image processing methods and more recently, using convolutional neural networks. However, 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.



(a) Different camera positions (b) When is the food plate visible?

Fig. 1. Capturing Food Images vs. Smartwatch Position

III. Feasibility Analysis

A. Study 1 - Preliminary Controlled-Study and Insights

In contrast to past work that focused mainly on eating gesture detection ([5], [10]), in this work, we additionally explore the plausibility of obtaining an image of the food being consumed through a watch's camera. Via a micro-study, we investigate whether this depends on the type of food or the on-watch placement of the camera.

Dataset Details: An extensive *micro study* was performed with 21 participants (8 females), belonging to 5 nationalities, employed in our university, covering a total of 135 eating episodes in the university's food court. An eating episode is defined as the period between the start of a meal (after the purchase) and consuming the last spoonful. 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 ground truth labeling). Separately, we also recruited 2 additional users to investigate the sensitivity of the results to the on-body location & orientation of the smartwatch camera. These users wore three distinct smartwatches, Samsung Gear 1, Samsung Gear 2 and Omate TrueSmart, each with the camera mounted in a distinct position. By varying the orientation on the wrist, we obtained 7 different camera positions, as illustrated in Figure 1(a).

Key Insights: Table I highlights some of the key parameters associated with the consumption of these food types. From these studies, we make the following key observations.

Duration of the Meal: From Table I we observe the wide variations in eating gestures for different food types. For items like rice and noodles, episodes lasted anywhere between 3.5 to 20 minutes, involving 13 to 54 separate hand-to-mouth gestures[1]. Among these food items, we also observed that: (a) sandwiches/fruits presented the least number of distinct hand-to-mouth gestures (as users often held items close to their mouth between successive bites), and (b) "noodle soup" had high variability in the number of hand-to-mouth gestures mainly due to the use of forks vs. chopsticks (use of chopsticks, generally leads to higher number of gestures).

Possibility of Image Capture & Orientation Sensitivity: From the images captured during the 135 episodes, after manually annotating, 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

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

TABLE I
KEY RESULTS FROM STUDY 1 (MICRO STUDIES)

Food Item	Preferred Eating Modality	Number of Episodes	Completion Time (sec)			Hand to Mouth Gestures			Percentage of Episodes with Useful Frames
			min	max	avg	min	max	avg	
Rice ≈100gms with 2 vegs	fork and spoon	66	211	1140	568	22	54	33.5	95.5%
Sandwich (bread slices/burger & fries)	hand	20	255	363	299	6	35	14.4	65%
Pasta / Soupy noodles	fork / chopstick	29	234	771	459	13	35	27.3	86.2%
Fruits ≈15 pieces	fruit stick	20	51	387	183	7	23	13.5	70%

all food types, except for sandwiches and fruits (in situations where the user never put the food 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. We found that the food plate is 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, Figure 1(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). Our studies help establish guidelines on where the image sensor must be placed on wrist-worn wearables, to support such diet capture.

IV. SYSTEM ARCHITECTURE

Figure 2 provides the high-level workflow of our automated food journaling system – *Annapurna*. *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). Broadly, the components on the smartwatch identify the intermittent *eating episodes* during the day, and then trigger the camera to capture *likely* images of the food being 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.

V. GESTURE RECOGNITION & TRIGGERING THE CAMERA

To detect individual eating gestures in intermittent eating periods, *Annapurna* employs a two-level classifier. The *Level1 Classifier*, uses only the accelerometer-based features, defined over a large (2.5 sec) time window. This classifier identifies suspected hand-to-mouth gestures with lower precision but more importantly, with a low energy overhead. The *Level2*

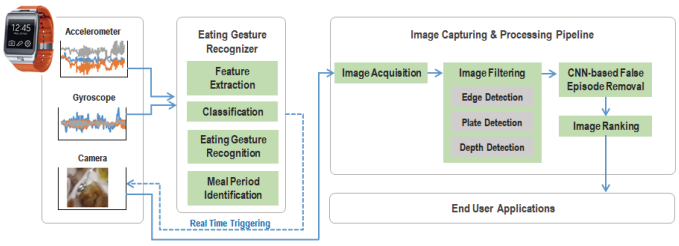


Fig. 2. System Overview

Classifier, when triggered by the *Level1 Classifier*, applies a subsequent set of (accel + gyro)-based features defined over a smaller (0.5 sec) window. This stage consumes more energy, but it is more precise in identifying hand-to-mouth gestures. Based on a 5 second window output of the *Level2 classifier*, an eating gesture is determined. If 3 hand-to-mouth gestures are identified within a window w_E (60 seconds in current implementation), then *Annapurna* determines that an eating episode has started. The details of the working of the gesture recognizer as described above has been published in [7], and it was found to achieve a precision of 93% and recall of 96.5%.

In this paper, we have used additional heuristics to reduce the energy consumption of the gesture recognizer.

- If *Annapurna* detects that the hand has been at rest (determined by the Variance of the accelerometer readings) for n_{rest} ($n_{rest} = 120$ seconds in current implementation) seconds, it assumes that the user is resting, and thus switches off all sensors for the next $n_{rest}/2$ seconds.
- As it is unlikely that a person will have back-to-back meals, *Annapurna* turns off all sensors for n_{b2b} seconds (currently it is 600 seconds) once it detects the end of an eating episode.
- If the smartwatch detects certain activities – e.g. typing on keyboard, walking, running, etc., it turns off all sensors for n_{act} seconds ($n_{act} = 180$ seconds in current implementation), as it is highly unlikely for a user to have a meal while concurrently performing these activities.
- If the *Level 2* classifier does not observe n_{Min} hand-to-mouth gestures in w_{Min} seconds($n_{Min} = 3$, $w_{Min} = 300$ sec), then it switches back to the *Level1 Classifier*, thereby reducing the energy overhead of running a gyroscope.

In addition to the above mentioned energy conservation heuristics, the final *Annapurna* system also utilizes human-in-the-loop principles to conserve energy and ensure accuracy. Whenever the commencement of an eating episode in determined, a pop up appears on the smartwatch to confirm if the

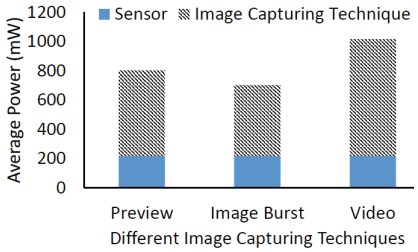


Fig. 3. Power Consumption for Image Capturing

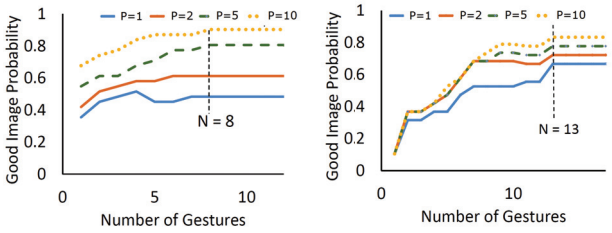


Fig. 4. Comparison of likelihood of acquiring a good image by the *Till-gesture* (left) and the *In-gesture* (right) strategies

individual is eating. If the user responds negatively to the pop up, *Annapurna* stops the sensors and image capturing for the subsequent 300 seconds. If the user responds 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 (details in Section VIII).

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*).

VI. Image Capturing Technique

Once the camera is triggered by the gesture-recognizer, the natural choice of recording images is either as a continuous video or as a burst of still images. Given the limited battery capacity of smartwatches, continuous video recording is not feasible. By measuring the battery drain for continuous video capture, we found that the battery drains from 100% to 10% in $\approx$ 80 minutes. The alternative strategy of capturing a still image has two issues: (i) from triggering the camera to capturing a single image, the latency is $\approx$ 900 msec, and (ii) as the number of images captured was lower, the possibility of capturing an usable food plate image is extremely low.

We then investigated the possibility of capturing *Preview frames*. Android exposes APIs which allows developers to grab the preview displayed on the screen. This preview refreshes at a high rate (more than 20 fps in the Galaxy Gear 1), thus solving the latency issue that exists in single image mode. While of slightly 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. Figure 3 shows the power consumption (measured using the Monsoon Power monitoring tool [15]) for different approaches (along with inertial sensing) as compared to the baseline. From the figure, we see that the *Burst* mode consumes the least power, while the *Preview* mode consumed only marginally higher power. However, 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 has to be answered is – after the camera is triggered, how long should it be kept on? By keeping the camera on for a longer duration within an eating episode, we are more likely to capture effective images for the food journal, but this will consume more energy. 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 from among all images captured till the N^{th} gesture, for each of the 135 eating episodes in Study 1 (details of the ranking algorithm is explained in Section VII). If we considered the top 10 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 approx. 135 seconds. 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 seconds. Even though the second approach kept the camera sensor ON for a shorter time duration, *Annapurna* adopted the former approach (*Till-gesture* strategy) because of the complexities that arise due to latency involving triggering of the camera. Evaluation of both strategies for different values of p and N is shown in Figure 4.

VII. Image Filtering

Our studies showed that many 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 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.

A. Lightweight Pre-processing on the Phone

Images acquired by the smartwatch are transferred to the paired smartphone via Bluetooth. To prevent *all* the images from being transferred to the server, *Annapurna* implements a lightweight pre-processor on the phone. The pre-processor includes (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 is blocked); the face detector eliminates images with visible human face. The edge

TABLE II
EFFECTIVENESS OF IMAGE FILTERING

Filtering Step (921617 Images)	Device	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

detector filters images based on the observation that when a clear image is captured, the image has a large number of edges, whereas blurry images have a smaller number of edges (due to subtractive mixing caused by motion blur).

By performing these simple 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. Based on power measurements performed, we found that the edge detection step consumes 39.5 mJ of energy to process an image, which is not very high.

B. Detailed Processing on the Server

The remaining images are transferred to the server, where they are processed to identify the relevant images.

Determine plate boundary Based on our assumption that the food consumed in served on a plate that has a regular, convex shape (either rectangular or circular), we use standard boundary detection methods available in $OpenCV$ to detect the bounding shape that covers the whole plate of food.

Eliminating non-food images and neighbor's food images using a depth map and CNN 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). The dominant SURF features [16] 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. 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. To further ensure that the image is indeed that of a food item, we then invoke the API provided by Clarifai inc. [17]. This API utilises 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 II shows the the average fraction of images that were eliminated at each step on our collected Study 1 corpus.

VIII. IN-THE-WILD EVALUATION

We next explored the feasibility of *Annapurna* as an automated food journal, that tracks a user's eating episodes throughout the day (with the user performing their natural daily activities). To this end, we recruited participants to wear the smartwatch continuously throughout the day. We eventually performed four distinct in-the-wild studies, with 7 participants in the first study, 6 in the second, 4 in the third, and 7 participants in the fourth study. The first two studies progressively-refined *Annapurna's* gesture recogniser. In this section, we provide details of in-the-wild Study 2, which was performed using the stablized gesture recognition model from [7], and Study 3, which was performed with an improved app including the heuristics based enhancements (described in Section V).

A. Study 2: In-the-Wild Food Journaling

A total of 4 participants were recruited for Study 3 and were provided with a smartwatch and a smartphone. 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 had to validate the images uploaded to *Annapurna* Web Journal at the end of the day.

Table III provides the details of the performance of *Annapurna* for the 4 participants. In the table, TP (true positive) indicates eating episodes that were correctly identified and displayed to the user, FP (false positive) refers to episodes that our system falsely inferred as eating, and FN indicates eating episodes that we missed. 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, but the images acquired from 7 of these cases were rejected by the image filters.

B. Study 2: User Feedback

At the end of Study 2, we surveyed the participants (from Study 2 and also the studies in [7]) about the scope of improvement for the system. 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). Further, we also realized that some simple heuristics (described in section V) may help improve the energy efficiency and accuracy. Therefore we performed a final in-the-wild user study with the above enhancements.

C. Study 3: Enhanced Food Journaling System

For Study 3, 7 participants (3 females) were recruited. However, 2 participants had issues with the app running on their watch, and hence could not be effectively included in the study. A total of 26 person days of data was collected from the 5 participants, which resulted in 152.3 hours of data collection. During the 152.3 hours, 51 meals were consumed by the participants while they wore the smartwatch. 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

TABLE III
PERFORMANCE OF *Annapurna* IN STUDY 2 & STUDY 3.

		Study 2				Study 3					Overall
Grnd 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 (hrs)	-	-	-	-	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-journal	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	All Sensors OFF (hrs)					11.7	5.4	8.3	3.9	8	37.3
	Accel ON (hrs)					13.6	14.9	19.1	14.9	18.6	81.2
	Accel + Gyro ON (hrs)		N/A			7.8	5.3	4.6	6.8	6.5	31
	Camera ON (hrs)					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-journal*: False positives in the final journal after image filtering, *FN*: False negative (Meals missed by App), *FNs Camera*: FN due to Camera triggering failure. Note: Participant # 4 participated in both Study 2and Study 3.

positive (FP) episodes were detected by the gesture recognition module, of which 34 were eliminated by the image filtering module (FP-Journal= 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 $\approx$ 20 mW power, accel + gyro together consumed ≈ 150 mW power and the camera along with the inertial sensors consumed ≈ 800 mW power. In Study 3 we found that based on the simple heuristics, we could turn off all sensing for $\approx 24.5\%$ 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, which indicates that *Annapurna's* energy overhead is indeed manageable.

D. Overall Performance of Annapurna

From studies 2 and 3 we found that overall *Annapurna* could recall 95% meals 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 reduced the sensing time by 25%, while the app had expensive sensors (camera and gyroscope) disabled for approx. 80% of a typical day.

IX. CONCLUSION

We presented *Annapurna*, an automatic smartwatch-based food journaling system. Through a micro-study, we demonstrated that a smartwatch-embedded camera can indeed capture such food-related images, with the preview mode providing the right balance between energy efficiency and image relevance. Through image filtering and processing, we demonstrated that *Annapurna* could identify the correct images during an eating episode in over 80% cases. Finally, through multiple user studies we showed that *Annapurna* could achieve a precision and recall of 95% in naturalistic environments.

REFERENCES

[1] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen, "Image browsing, processing, and clustering for participatory sensing: lessons from a dietsense prototype," in *Proceedings of the 4th workshop on Embedded networked sensors*, 2007.

[2] G. O'Loughlin, S. J. Cullen, A. McGoldrick, S. O'Connor, R. Blain, S. O'Malley, and G. D. Warrington, "Using a wearable camera to increase the accuracy of dietary analysis," *American journal of preventive medicine*, vol. 44, no. 3, pp. 297–301, 2013.

[3] J. Liu, E. Johns, L. Atallah, C. Pettitt, B. Lo, G. Frost, and G.-Z. Yang, "An intelligent food-intake monitoring system using wearable sensors," in *Wearable and Implantable Body Sensor Networks (BSN), 2012 Ninth International Conference on*. IEEE, 2012.

[4] K. Yatani and K. N. Truong, "Bodyscope: a wearable acoustic sensor for activity recognition," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. New York, NY, USA: ACM, 2012.

[5] E. Thomaz, I. Essa, and 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*, 2015.

[6] S. Sen, V. Subbaraju, A. Misra, R. Balan, and Y. Lee, "The case for smartwatch-based diet monitoring," in *Pervasive Computing and Communication Workshops (PerCom Workshops), IEEE*, 2015.

[7] S. Sen, V. Subbaraju, A. Misra, R. Balan, and Y. Lee, "Experiences in building a real-world eating recogniser," in *Proceedings of the 4th International on Workshop on Physical Analytics*.

[8] P. Kelly, S. J. Marshall, H. Badland, J. Kerr, M. Oliver, A. R. Doherty, and C. Foster, "An ethical framework for automated, wearable cameras in health behavior research," *American journal of preventive medicine*, vol. 44, no. 3, pp. 314–319, 2013.

[9] T. Y.-H. Chen, L. Ravindranath, S. Deng, P. Bahl, and H. Balakrishnan, "Glimpse: Continuous, real-time object recognition on mobile devices," in *ACM Conference on Embedded Networked Sensor Systems*, 2015.

[10] O. Amft, H. Junker, and G. Troster, "Detection of eating and drinking arm gestures using inertial body-worn sensors," in *Wearable Computers. Proceedings. Ninth IEEE International Symposium on*. IEEE, 2005.

[11] Y. Dong, J. Scisco, M. Wilson, E. Muth, and A. Hoover, "Detecting periods of eating during free-living by tracking wrist motion," *Biomedical and Health Informatics, IEEE Journal of*, vol. 18, no. 4, 2014.

[12] "Myfitnesspal: Free calorie counter, diet and exercise journal," http://myfitnesspal.com/, accessed: 2017-02-15.

[13] B. L. Heitmann and L. Lissner, "Dietary underreporting by obese individuals–is it specific or non-specific?" *Bmj*, vol. 311, no. 7011, pp. 986–989, 1995.

[14] Y. Kawano and K. Yanai, "Foodcam: A real-time food recognition system on a smartphone," *Multimedia Tools and Applications*, vol. 74, no. 14, pp. 5263–5287, 2015.

[15] "Monsoon power monitor," https://www.msoon.com/LabEquipment/PowerMonitor/, accessed: 2017-02-15.

[16] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[17] "Clarifai inc: Image and video reognition api," http://www.clarifai.com/, accessed: 2017-02-15.